Research Article

Low-Latency Wireless Data Transfer for Motion Control

Jeroen de Boeij,¹ Maarten Haazen,² Peter Smulders,² and Elena Lomonova¹

¹ *Electromechanics and Power Electronics Group, Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands*

² *Radiocommunications Group, Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands*

Correspondence should be addressed to Jeroen de Boeij, jeroen@deboeij.com

Received 14 January 2009; Accepted 2 July 2009

Recommended by Benoit Boulet

This paper discusses a new approach for wireless motion control. Existing wireless techniques suffer from large closed loop delays of several milliseconds, which is unacceptable in precision motion systems. These large delays are mainly caused by the protocol used, since these are optimized for transferring large amounts of data, not to minimize transmission delay. A new protocol and wireless system are proposed that reduce the closed loop transmission delay to less than 300 microseconds. The system has been verified in a real control system, and measurements show the performance, which is more than ten times better than existing techniques.

Copyright © 2009 Jeroen de Boeij et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

In high-tech mechatronic systems a cable slab to the moving part of the machine is a source of disturbances. It introduces friction, stiffness, and damping and limits the stroke of the mover. In addition, the cables break after a certain number of movements due to the strains induced by the motion, resulting in machine failures. The application of wireless motion control is most promising in the semiconductor industry, where increasingly higher throughput and accuracy are demanded, for example, lithography, component placement, surface mounting, and wire-bonding.

When using air bearings or magnetic bearings, the cable slab is the only physical connection to the fixed world. The cable slabs are necessary to provide power to the mover and exchange information between the moving part and the controller, which is located on the fixed world. To omit a cable slab, both the energy transfer and information exchange should be made wireless. Contactless energy transfer is possible by means of an inductive coupling. Many wireless techniques exist that can reliably transmit data, such as WLAN TCP/IP, WLAN UDP/IP, Bluetooth, and Zigbee [1–3]. These protocols, based on radio communication in the 2.4 GHz ISM band, are designed to transmit large amounts of data and are optimized for high data throughput. With

the exception of WLAN UDP/IP all protocols guarantee that the data arrives correctly at the receiving side, by means of packet retransmission. Since the time between sending and receiving is not critical for sending large data files such as website information, music, and movies, the delay is in the order of several milliseconds for WLAN UDP/IP and up to 20 milliseconds for the other protocols [4–7]. Although a theoretical transmission time as low as several hundreds of microseconds is estimated for WLAN UDP/IP in [7] and for the latest 802.11e MAC level using Immediate Block Ack or Direct Link Protocol [8], measured data in [4] shows that the projected best case performance is not met in real situations.

Moreover, this delay is not constant but it has considerable variation or jitter [4, 6]. In some applications such a delay is no problem, like sensor systems to monitor temperatures [9] in buildings; however, for high-precision machines delays in the order of milliseconds are unacceptable for fast servo loops.

A lot of work is done to deal with this delay with observers and predictors [10–14] and to prove the stability of systems suffering from variable transmission delays [15, 16]. There is not much literature about wireless control applications. In one case an inverted pendulum is controlled using WLAN UDP/IP with a closed loop delay of 3.6 milliseconds

[4], with a controller sample time between 167 and 250 Hz. The performance is mediocre, with angle errors of more than 20 degrees, although no observer or compensation is used for the delay. In another case, a similar plant is controlled using bluetooth [6] with a transmission delay of 17 milliseconds, which shows similar performance to the results presented in [4] and even instability if the delay compensation is turned off.

All these contributions use the standard protocols that are not developed for data transmission with a minimal delay. Moreover, using a wireless data transmission with delay in combination with observers and predictors will never work as well as removing, or greatly reducing, the delay itself.

The starting point of this paper is that none of the currently existing wireless protocols are suitable in motion control applications. Therefore, a new protocol, specially developed to minimize delays in the transmission, is investigated in this paper. Since all protocols are strongly correlated with the hardware [1, 2] and usually low-level parts of the protocol such as MAC are embedded in the hardware, the hardware implementation of such a protocol is important as well. The requirements of such a real-time wireless link, software, and hardware implementation of the protocol and test results of the prototype are discussed in this paper. Finally, the behavior and stability of the wireless link in a real motion system are shown.

2. Requirements for a Real-Time Wireless Link

First, the requirements for a real-time wireless link are discussed, which are then compared with the characteristics of currently existing wireless techniques. Finally, a choice is made for the wireless channel and the protocol. There are three characteristics that are very important for a real-time wireless link.

2.1. Accuracy. In high-precision systems position is measured in micrometers or even nanometers. For a linear motor with a stroke of one meter and an accuracy of $1\text{ }\mu\text{m}$, this implies a precision of 20 bits. This prohibits the use of fully analog techniques, since it will be difficult if not impossible to maintain the precision. Therefore, the information should be sent digitally using data packets.

2.2. Delay. A major issue is the definition of real-time, which has different meaning, depending on the application. The transmission delay or latency that is acceptable in the application usually defines whether something is real-time or not. In control, closed loop delay is one of the characteristics that determines the performance (i.e., bandwidth) of the controlled motion system. An ideal plant is considered, which consists of a mass ($P(s) = 1/ms^2$) with 2 milliseconds output delay. To investigate the influence of delay on the control bandwidth a controller with a stabilizing lead-lag-filter with a zero at one-third of the bandwidth and a pole at three times the bandwidth is assumed. The bode diagram of the plant and the controller is shown in Figure 1. This results

in approximately 50° phase lead at the control bandwidth [17] and with a phase margin of 30° . The phase margin is shown in Figure 2. The following holds for the maximum bandwidth f_{bw} for a given closed loop delay δ_{cl} :

$$f_{bw} = \frac{50 - 30}{360\delta_{cl}}. \quad (1)$$

In a plant the delay due to the sensor delay, actuator delay, and computing time is typically 1 millisecond; so adding another 1 millisecond by the wireless links reduces the maximum controller bandwidth from 56 Hz to 28 Hz. Since there are two wireless links in the loop, the maximum permissible delay for one link is only 500 microseconds.

In high-precision motion systems, with closed loop bandwidth requirements up to several hundreds of Hz, this delay is still unacceptable so a further reduction of the delay is desirable. However, the requirement of 500 microseconds is used as a start to benchmark the suitability of existing wireless techniques for wireless motion control.

Delay can be split into two parts.

(1) *Static Delay.* This type of delay is always the same and is caused by processing of the data in the protocol and the limited data rate. This can be kept small by using a simple protocol with small data packets and a wireless channel with a high data rate.

(2) *Stochastic Delay.* This is delay that is not constant but varies in time, mainly due to the retransmission of wrongly received data. The variation of the total delay is called jitter and is caused by this stochastic delay, which is purely related to the protocol. Moreover, retransmission of data is not useful for control applications, since it is better to wait for the next data packet than to waste time by retransmitting the old one. Other phenomena such as routing also contribute to stochastic delay.

2.3. Data Rate. The data rate of the wireless channel directly influences the delay: if the data rate is doubled than the time needed to transmit, a packet through the wireless channel will halve. The data rate strongly depends on the frequency of the electromagnetic wave used to transmit the data and the modulation technique; for example, in optical systems data rates of Gbps are possible, and the currently available WLAN operates at 54 Mbps.

The data rate necessary to realize a wireless motion control system needs to be estimated. Each position measurement is about 20 bits accurate ($1\text{ }\mu\text{m}$ resolution for a 1 m stroke). The sample frequency f_s of the controller is 1 kHz, although higher frequencies are increasingly common (up to 20 kHz in nm stages). A significant number of actuators are single phase or 3-phase motors, which require one or two current setpoints, respectively, estimated at 12 bit per setpoint. If the commutation of the 3-phase motor is done locally, only one current amplitude setpoint is necessary. Finally, several enable and status bits are necessary to operate a motor. The manipulator consists of three 3-phase motors, and from this discussion it is

FIGURE 1: Bode plot of an ideal plant and a controller.

FIGURE 2: Nyquist plot of the open loop transfer function PC .

assumed that 64 bits of data per packet is sufficient to transmit the necessary information. However, in wireless data transfer additional bits are necessary to encode the data into packets. Assuming that about two times the number of data bits is required and that 64 bits of data is sent at 2 kHz, the required data rate is finally estimated at 256 kbps.

3. Characteristics of Existing Wireless Techniques and Protocols

Several wireless techniques are currently available that send digital information, such as WLAN, Bluetooth, and Zigbee [1–3]. Usually these protocols have strong links to the hardware implementation, but in principle any medium could be used to transmit the bits. All of these techniques operate in the 2.4 GHz ISM frequency band and their characteristics are listed in Table 1. From this table it is clear that all techniques have sufficient bandwidth. However, none of the techniques is able to meet the minimum delay specification of 500 microseconds. Another interesting characteristic is the difference between WLAN TCP/IP and WLAN UDP/IP. The delay for TCP/IP is about ten times higher than for UDP/IP. This is caused by the fact that UDP/IP is designed for streaming video transfer over internet and does not use packet retransmission; that is, wrongly received packets are not sent again. To meet the delay specification of 500 microseconds, it is necessary to develop a new protocol.

4. Custom Protocol

The delay requirement of the wireless system can only be met by means of a custom protocol with small packets that only consist of data and a minimum of additional overhead. Packet retransmission is omitted, since it is undesired in control applications; that is, only the most recent data is relevant. Since wireless motion control is confined to a

TABLE 1: Characteristics of several wireless communication techniques.

Communication technique	Bandwidth	Minimum delay
Bluetooth	1 Mbps [2]	7 ms [5, 6]
IEEE 802.11b WLAN (TCP/IP)	11 Mbps [1]	20–25 ms [5]
IEEE 802.11b WLAN (UDP/IP)	11 Mbps [1]	1.78 ms [4]
Zigbee	250 Kbps [3]	15 ms [18]

machine or system, routing is omitted, since the inclusion of such a system needs to be incorporated from the start of the design. A smart use of channels and the limited range of transmission enables routing on a hardware level.

The wireless link must transmit 64 bits of data in a packet. How this data is acquired will be discussed later in this section. As discussed previously, the packet must be as small as possible to minimize the delay of the transmission. However, some overhead is absolutely necessary to ensure the correct transmission of the data.

(1) *Start Bits*. The beginning of the packet is marked with sixteen start bits that are the same for every packet. This ensures that the beginning of each packet is recognized by the receiver.

(2) *CRC-16*. To verify the validity of the data bits a 16-bit key is sent with the data to perform a cyclic redundancy check (CRC). CRC is a polynomial code, which means that bit strings are treated as a representations of polynomials with coefficients of 0 and 1 only. The CRC-algorithm is an error detection code and can be put in the following general form:

$$M(x) \cdot x^n = Q(x) \cdot K(x) + R(x). \quad (2)$$

Here $M(x)$ is the original message polynomial, 64 bits of data, and $K(x)$ is the key polynomial, with degree n . The bits of $M(x) \cdot x^n$ are the original message with n zeros appended at the end. $R(x)$ is the remainder polynomial, which is the CRC. In communication, the sender attaches the n bits of R after the original message bits of M and sends it to the receiver. The receiver takes M and R and checks whether $M(x) \cdot x^n - R(x)$ is divisible by $K(x)$. If it is, then the receiver assumes that the received message bits are correct. From measurements at different data rates, a CRC-16-IBM ($n = 16$) code was found to be required to detect all errors. CRC-16-IBM is also used in USB. The key polynomial is $1 + x^2 + x^{15} + x^{16}$.

(3) *8b10b Encode*. The data transmission is asynchronous, which implies that the transmitter and the receiver do not operate with the same clock. Therefore, the clock rate of the transmission needs to be recovered. In order to be able to do so, there must be sufficient transitions between zeros and ones in the bitstream of the transmission. The 8b10b encoding guarantees that there are no more than three consecutive zeros or ones in the bitstream to accommodate the clock-recovery. The number of bits however increases with 25%.

(4) *Error Detection Bits*. Since the packet is very small, it is expected that a maximum of 1 bit per packet will be wrong. A Hamming code of 7 bits is added to the packet, which can correct a single erroneous bit and detect two.

Based on this structure of the packet, the total packet size n_p will be

$$16 \text{ Start} + 1.25 \times (64 \text{ Data} + 16 \text{ CRC}) + 7 \text{ Hamming} \\ = 123 \text{ bits.} \quad (3)$$

As can be concluded from the description of the packet, its structure is very basic and stripped of anything that is not absolutely necessary. Although it seems simple, such an approach is not documented in literature but offers great advantages for real-time motion control by minimizing the delay.

5. Hardware

The implementation of such a protocol requires a transceiver without any embedded protocol. A lot of transceiver chips for standard techniques (WLAN, bluetooth) have parts of the protocol embedded in the hardware and are therefore unsuitable for the development of custom protocols. Unfortunately, any protocol is strongly related to its hardware implementation; so this is also included in the discussion.

Optical techniques such as laser are not considered, since they require line-of-sight (LOS). This is hard to achieve for a motion system that has a long-stroke movement in the three-dimensional space. The advantage of optical systems is a high bandwidth and low bit-error-rates (BERs), but the LOS requirement makes it more difficult to implement. In a system with a purely rotating or linear movement, optical systems are more feasible. Therefore, radio waves are the best choice for the wireless data transfer, in particular the unlicensed 2.4 GHz ISM band.

There are not many commercially available radio transceivers that have no embedded protocol, and so the choice is limited. Moreover, RF design requires specific skills and it is time consuming. To overcome these issues a ready-to-use RF starters kit (ML2724SK-02 from Micro-Linear [19]) is used that is based on a 1.5 Mbps FSK transceiver. If the transceiver is used as a transmitter, it converts zeros and ones on the input to an RF antenna signal. When the transceiver is used as receiver, the RF antenna signal is converted back to zeros and ones on the output. Due to filtering of the RF signal, variation of the time of each bit (jitter), and noise, not all bits are transmitted correctly. In Figure 3 the bit-error-rate (BER) is measured at different data rates. Since the packet size is 123 bits, data rates higher than 1.5 Mbps are unacceptable due to the high BER. Moreover, this measurement is done with only one link. In the final setup the transmission will be full-duplex, that is, simultaneously transmitting and receiving in two directions, and so there will be two transmitters and two receivers, which will interfere with each other. Therefore, it is expected that the BER in the full-duplex system will be worse than this measurement.

FIGURE 3: BER measurement of ML2724SK-02 transceiver.

High-gain, directional antennas are used to limit the interference between the channels and the influence of other 2.4GHz transmitters in the lab such as the WLAN router. Since the transmitter and receiver will only be one or two meters apart, the signal from these antennas is very strong within the workspace of the wireless link. Basically, the high-gain antennas ensure that any other signal at that channel is overpowered. WLAN routers or bluetooth devices in the same area cannot use the two channels occupied by the wireless motion system, but their architecture is suitable to deal with that.

The transceiver has a adjustable amplifier to change the RF signal strength from the antenna. To prevent clipping of the signal, the gain is automatically adjusted by a microcontroller. This hardware choice of the wireless communication channel is by no means optimal, but given the commercial availability it is the best option.

The protocol itself is implemented on an FPGA (Xilinx Spartan IIE [20]). An FPGA is chosen since it allows parallel execution of code. Moreover, integer arithmetic is very fast due to the use of interconnected logic gates. In addition, it can execute several threads at different clock rates, making the FPGA a flexible development platform. The communication between the FPGA and the outside world (sensors, amplifiers and DSP) is done by means of digital IO and AD/DA conversion.

5.1. Transmitter Protocol. Several operations are performed by the FPGA to send data by one of the wireless transmitters (Figure 4). First, the sending of the start bits to the parallel-to-serial converter is started. While these bits are being sent, the DA converters are triggered and the data is acquired from the AD converters or the incremental decoder. In a single clock cycle, a CRC-16 is calculated from this data, and the 64 bits and CRC bits together are 8b10b encoded, followed by the calculation of the Hamming correction parity bits. This data is added to the packet behind the start bits, just before all start bits are sent. Finally, the rest of the packet is put into

the parallel-to-serial converter, driving the transmitter with this serial data. The timing of this A/D trigger assures that the most recent data is sent. The parallel-to-serial converter triggers the transmission of the next packet.

5.2. Receiver Protocol. To retrieve the data correctly from the receiver, and due to the asynchronous reception, a 16x oversampling and clock recovery algorithm has been implemented. The clock recovery is required because the oscillator (24.756 MHz) of the transmitter will never have exactly the same oscillation frequency as the oscillator at the receiver. Oversampling is also a necessity caused by each received bit period deviating from the average value (jitter). A Moore state machine is used as the oversampling algorithm (Figure 5).

When 9 consecutive samples are recognized with the same value, it is recognized as a valid input-bit. This algorithm also automatically recovers the clock from the data; hence it is known as a DCR (Digital Clock Recovery) circuit. This oversampling algorithm also compensates for the jitter (variations in bit duration). However, the algorithm only works properly when sufficient logical transitions are present in the data. To ensure sufficient logical transitions, an 8b10b encode has been applied.

While each bit is being received using the oversampling algorithm, each bit is concatenated into a 16-bit register like a sliding window. The data in the register is compared with the start bits. When the data in the shift register is equal to the start bits, a signal is generated which will trigger the data processing of the rest of the packet. The rest of the packet is put in an array of 107 bits. This is followed by a Hamming correction decode, which is able to correct a single erroneous bit. Consequentially, this data is 8b10b decoded, which leaves 80 bits. For the first 64 bits of this array a CRC is calculated, which is compared with the last 16 bits. Equality of the CRC's means that the first 64 data bits are valid and are thus sent to the output port. When a whole packet has been received (valid or not), a signal will trigger the search for the start bits of the following packet. False packets are ignored, due to the useless nature of data retransmission in motion control, and the previous valid data is hold at the output port. The entire protocol is visualized in Figure 6.

5.3. Expected Performance. Using the packet size n_p and the data rate d_r of the data transmission, the delay in a single wireless link δ_{sl} is estimated:

$$\delta_{sl} = \frac{3n_p}{2d_r}, \quad (4)$$

where the factor 3/2 is added to take the sample and holds characteristic of the transmission into account. Besides the delay due to data rate limit, there is additional delay that is introduced by the AD/DA conversions and the buffers in the digital IO. In addition, the sample frequency is calculated by

$$f_s = \frac{d_r}{n_p}. \quad (5)$$

Since a wireless control system consists of two links the additional closed loop delay added by the two links is double

FIGURE 4: Schematic of FPGA transmitter protocol.

FIGURE 5: Moore state machine for the oversampling algorithm.

TABLE 2: Delay and sample frequency for different data rates.

d_r (Mbps)	δ_{wl} (μs)	f_s (kHz)
0.768	480	6.24
1.024	360	8.33
1.280	288	10.4
1.536	240	12.5

the single link delay. In Table 2 the additional closed loop delay δ_{wl} and sample frequency f_s are listed for different data rates d_r , considering only the delay due to the limited data rate.

6. Identification of the Wireless Control System

To test these techniques a contactless planar actuator with manipulator is studied as shown in Figure 7. It consists of a moving magnet planar actuator with 6 DOF combining electromagnetic suspension and propulsion with a manipulator on top. The energy transfer is done by means of an inductive coupling between the fixed primary coils and the moving secondary coil.

The manipulator consists of a translating beam with two linear motors and an arm driven by a rotating motor. All three motors are coggingless and are equipped with encoders

($1\mu m$ and $40\mu rad$ resolution, resp.). The manipulator on top of the floating platform needs to be controlled in synchronization with the magnetic suspension. Furthermore, in order to ensure accurate positioning of the tool center point (TCP) in 3D space, it is necessary to know the position of the platform and the position of the TCP of the manipulator with respect to the platform in real-time. Finally, the movement of the magnetic suspension and of the manipulator will influence each other, which requires feed-forward compensation. The realization of these functions requires a real-time wireless communication link between the ground controller of the magnetic suspension and the floating platform. Ideally, the controller of the manipulator is located on fixed world, together with the controller of the magnetic suspension, which requires the entire control loop of the manipulator to be wireless. The proposed real-time wireless system is implemented to realize a wireless control loop. Measurements concerning the packet loss have been executed to evaluate performance of communication, and delay measurements have been done to determine the effects of the wireless communication compared to wired solutions. A position error measurement determines the performance regarding control.

6.1. Packet Loss. In Figure 8, the packet loss ratio, which is the ratio between the number of noncorrectable erroneous packets and the total number of received packets, has been measured for four different data rates. In addition the ratio between the number of corrected packets, which had a single bit error that is corrected by the Hamming code, and the total number of received packets is plotted. The results show the difference in performance between one active link and two active links, caused by the interference between the links. The figure shows that up to about half of the packets which contained an error are corrected. Consideration should be made what is preferred for a specific plant: a rather large packet loss with the advantage of a very short delay, or a smaller packet loss ratio with a larger delay.

FIGURE 6: Schematic of FPGA receiver protocol.

FIGURE 7: Contactless planar actuator with manipulator.

6.2. Transfer Function and Delay. To test the wireless control system, a closed loop wireless control system has been built, in which the arm of the manipulator, driven by a three-phase synchronous rotary motor, is controlled. The controller DSP (dSpace DS1103) generates analog voltage signals, representing the motor current references, which are sent over the first wireless link to the input of the amplifier of the motor. The position of the arm is measured by means of an incremental encoder. The signals from this sensor are decoded to a 16-bit position, which is sent over the second wireless link and is read by the controller DSP using a 16-bit digital IO. The transfer functions of the setup at different data rates have been measured using the Siglab spectrum analyzer. These “wireless” transfer functions are compared with the transfer function of the same control system without the wireless communication (wired). Figure 9 shows that the wireless system has hardly any effect on the gain of the system. However, at high frequencies the gain decreases

FIGURE 8: Measured packet loss ratio and packet correction ratio.

with respect to the wired system, which is related to the fact that the commutation is done by the controller. The additional delay at this frequency influences the accuracy of the commutation, and an error in the commutation results in a lower gain. This can be improved by implementing the commutation on the FPGA and only sending the current amplitude over the wireless link.

The phase shift increases when the data rate decreases, which is consistent with the extra delay involved. From the phase shift in the transfer functions, the delay is calculated for each data rate. These delays are variable as shown by

FIGURE 9: Measured transfer functions of rotary motor with wireless link at different data rates.

FIGURE 10: Measured additional closed loop delay probability density at different data rates.

the probability density distributions in Figure 10, which is caused by the absence of synchronization between controller DSP and FPGAs. These difference between the maximum and minimum delay for each data rate is approximately 130 microseconds, which is significant; so synchronization should be a main issue in future improvements. The mean

FIGURE 11: Nyquist plot of wired and wireless open loop transfer function PC .

delays are rather consistent with the estimated delays, but slightly higher due to the A/D conversion times, buffers, and unsynchronized operation of the wireless system with the controller DSP. The variable delay can be reduced by synchronizing the wireless link and the controller, that is, start sending a packet with current setpoints directly after the controller finished calculating the setpoint values and transmit the latest position data just before the controller starts with the next sample. At the highest data rate, the average of the extra closed loop delay is just 270 microseconds. This delay is about 15 times shorter compared with the WLAN 802.11b solution [4], while just a fraction of the data rate is used (1.536 Mbps against 11 Mbps). In all cases, the additional closed loop delay remains below the 1 millisecond specification.

7. Position Control

To determine the effects of the wireless communication on the control loop performance the position error of the arm in a wired closed loop system has been compared with the position error of the arm in the wireless closed loop systems. A PID controller is implemented that realizes a position loop bandwidth of 30 Hz, defined as the 0 dB crossing of the gain of the open loop transfer function PC . The phase margin is 46 degrees and the gain margin is 15 dB, if the delay of the wireless link is omitted. Due to the added delay in the loop, the phase and gain margin of the wireless loops are less and depend on the data rate of the wireless link. In Figure 11 the Nyquist plot of the open loop transfer function of the system is shown for the wired case and for the wireless case at different data rates.

If a packet loss occurs, additional delay is induced in the loop which is equal to the sample time of the controller plus the time delay between sending and receiving a packet. In Figure 12, the Nyquist plot of the open loop transfer function is shown including two sequential packet losses. In this case, all systems are stable but the margins have been reduced to approximately 30 degrees and 10 dB, while the maximum

FIGURE 12: Nyquist plot of wired and wireless open loop transfer function PC with two consecutive packets lost.

gain of the sensitivity is approaching 6 dB. Now it is possible to evaluate how many consecutive packets should be lost in order to cross -1 in the Nyquist plot. Taking into account the packet loss ratio measured in Figure 8 and assuming a uniform distribution for the occurrence of packet loss, it can be calculated that instability will occur once every 1×10^{19} seconds in the worst case.

The position error of the arm is plotted in Figure 13 for different data rates. Because the FPGAs and the DSP run at different sample frequencies (f_s , see Table 2 and 8000 Hz, resp.) and no synchronization is present in the wireless system, harmonics of the difference frequency are present in the position error of the wireless closed loop systems. At a data rate of 0.768 Mbps the error amplitude is significant higher than at higher data rates, because $d_r/n_p = 6244 \text{ Hz} < 8000 \text{ Hz}$. The large packet loss at 1.536 Mbps introduces significant spikes on the position error. These readings are confirmed by the maximum position error. Table 3 shows the maximum position error in milliradians and the percentage of errors larger than 2%. From Figure 13 and Table 3, it is concluded that the wireless control system performs best at a data rate of 1.280 Mbps. These results show a significant improvement compared to the inverted pendulum wireless control loops discussed in [4, 6], which show errors in the range of 10 degrees (0.2 rad) or worse.

The error in the wireless system suffers from a lot of chattering, which is not visible in the wired system. The error of the wired system is smooth, which indicates that it is related to unmodeled dynamics. The chattering on the error of the wireless system reduces significantly when the velocity of the arm is zero. It is likely that this effect is related to the commutation. If the velocity is close to zero, the angle of the rotor hardly changes during one sample time, and therefore, the phase currents that are commutated depending on the position do not change during the sample time. It suggests that the control behavior will be improved by implementing the commutation locally at the FPGA, while only sending the current amplitude command from the controller over the wireless link. In addition, the variable delay in the loop due to

FIGURE 13: Position error arm.

TABLE 3: Performance closed loop systems.

Control system	Maximum position error	% error > 2 mrad
Wired	2.0367 mrad	0.0079%
Wireless 0.768 Mb/s	4.0180 mrad	1.34%
Wireless 1.024 Mb/s	2.8486 mrad	0.0058%
Wireless 1.280 Mb/s	2.5788 mrad	0.0031%
Wireless 1.536 Mb/s	5.5010 mrad	0.24%

the wireless communication will have less effect on the performance when the velocity is small. This can be improved by synchronizing the wireless data transfer and the controller.

TABLE 4: Data allocation within the data packets.

Manipulator to controller	Controller to manipulator
3 positions of 20 bits each	3 current amplitudes of 16 bit each
1 bit motor 1 enabled	1 bit enable motor 1
1 bit motor 2 enabled	1 bit enable motor 2
1 bit motor 3 enabled	1 bit enable motor 3
	1 bit enable homing
63 bits	52 bits

8. Implementation of the Wireless System on the Manipulator

From the initial tests with a SISO system, the final wireless control system for the entire manipulator is derived. The first requirement is that all information should fit within the 64 bit of data in the packet. The commutation is done locally by the FPGA, which implies that the controller only needs to send three current amplitudes, one for each motor, that are converted to 3-phase current commands to the amplifiers. The allocation of data in the data packets is shown in Table 4. To save space on the manipulator platform, all communication is implemented on a single FPGA board. Since the same protocol and packets are used as in the tests with one motor, the performance of the system in terms of packet loss and delay will be the same.

9. Discussion

The problem with existing wireless communication methods is the large protocol overhead. These protocols are developed for the transfer of large files, and obviously, one does not care whether there is a delay of several milliseconds between the sending of a picture file and its delivery. In addition, a higher data rate does not imply a lower delay due to the protocol overhead. The approach presented in this paper demonstrates that a specific protocol is necessary to realize the low latencies required for motion control. The advantage of this approach is that an increase of the data rate does give an equal decrease of the delay. Since this work has focused on the wireless protocol, the wireless radio (2.4 GHz FSK transceiver) used is far from ideal and improvements can easily be made by using better transceivers (up to 400 Mbps UltraWideBand is now commercially available), infrared (up to 24 Mbps), or laser (up to several Gbps). The choice strongly depends on the application. Using a 54 Mbps transceiver that is used in WLAN in this system would result in an additional closed loop transmission delay of less than 10 microseconds. If transmission delays are reduced to these levels, redundancy can be incorporated, that is, sending multiple packets with the same position information in one sample, to guarantee a 100% reliability.

10. Conclusions

A new protocol and implementation for low-latency wireless communication is presented that is suitable for wireless

motion control and that overcomes the unsuitability of currently existing wireless communication protocols. The additional closed-loop delay is reduced to 300 microseconds which is more than 10 times smaller than other known methods. This low delay was realized with a low data rates up to 1.536 Mbps. Since the delay in this protocol is inversely proportional to the data rate, much lower delays and lower packet loss ratios can be easily obtained by optimizing the wireless channel. The behavior of a wireless motion system, consisting of an arm driven by a 3-phase ironless rotary motor and a PID controller with a 30 Hz bandwidth position loop, is shown. The stability of the system is demonstrated using Nyquist analysis of the open loop transfer function. The results show errors in the range of several millirads, and the behavior of the wireless motion control system can be further improved by adding synchronization between the wireless communication link and the position controller.

Acknowledgment

This research is sponsored by SenterNovem. SenterNovem is an agency of the Dutch Ministry of Economical Affairs.

References

- [1] IEEE, IEEE Standard 802.11-1997, <http://www.ieee.org/>.
- [2] Bluetooth specification, <http://www.bluetooth.com/>.
- [3] The Zigbee Alliance, The ZigBee Specification, <http://www.zigbee.org/>.
- [4] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Systems Magazine*, vol. 24, no. 3, pp. 58–71, 2004.
- [5] N. Golmie, "Wireless Personal Area Networks," IEEE P802.15-01/317r0, 2001.
- [6] J. Eker, A. Cervin, and A. Horjel, "Distributed wireless control using Bluetooth," in *Proceedings of the IFAC Conference on New Technologies for Computer Control*, Hong Kong, November 2001.
- [7] C. Hochel-Winter, "Industrial wireless LANs with Siemens SCALANCE W," Product Analysis, ComConsult Research, 2004.
- [8] Q. Ni, "Performance analysis and enhancements for IEEE 802.11e wireless networks," *IEEE Network*, vol. 19, no. 4, pp. 21–27, 2005.
- [9] M. Kintner-Meyer and R. Conant, "Opportunities of wireless sensors and controls for building operation," *Energy Engineering*, vol. 102, no. 5, pp. 27–48, 2005.
- [10] S. Behnke, A. Egorova, A. Gloye, R. Rojas, and M. Simon, "Predicting away robot control latency," in *Proceedings of the Robot Soccer World Cup VII (RoboCup '04)*, vol. 3020 of *Lecture Notes in Computer Science*, pp. 712–719, August 2004.
- [11] W. J. Kim, K. Ji, and A. Ambike, "Networked real-time control strategies dealing with stochastic time delays and packet losses," in *Proceedings of the American Control Conference*, vol. 1, pp. 621–626, Portland, Ore, USA, June 2005.
- [12] J. Nilsson, *Real-time control systems with delays*, Ph.D. thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.
- [13] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, 1998.

- [14] Y. Uchimura, "Wireless network based control with time varying delay," in *Proceedings of the 10th IEEE International Workshop on Advanced Motion Control (AMC '08)*, pp. 336–341, March 2008.
- [15] M. Tabbara, D. Nešić, and A. R. Teel, "Input-output stability of wireless networked control systems," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 209–214, Seville, Spain, December 2005.
- [16] W. Zhang, M. S. Branicky, and S. M. Philips, "Stability of networked control systems," *IEEE Control Systems Magazine*, pp. 84–99, 2001.
- [17] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, Reading, Mass, USA, 1994.
- [18] W. C. Craig, "Zigbee: Wireless Control That Simply Works," Zigbee Alliance ZigBee Alliance, 2004, <http://www.51zigbee.com/uploadfile/jh2.pdf>.
- [19] Microlinear, ML2724SK-02 High Power 2.4GHz Radio Starter kit Design.
- [20] Xilinx, Spartan IIE, http://www.xilinx.com/support/documentation/data_sheets/ds077.pdf.