

블루투스 Synchronous Connection Oriented Link 를 사용한 네트워크 제어 시스템

Bluetooth Synchronous Connection Oriented Link Usage in Networked Control Systems

Ulugbek Umirov, 박 정 일*
(Ulugbek Umirov¹ and Jung-II Park¹)
¹Yeungnam University

Abstract: In this paper the usage of Bluetooth in networked control systems is described. ACL links and commonly used serial port profile built on top of ACL links are analyzed and their problems such as unpredictable latency are discovered. SCO link packet scheduling, latency estimation and setup procedure are examined. SCO link is suggested as proper link for NCS, due to its low latency and low variance. Smith predictor use for latency compensation is described and its impact on control performance is estimated. A number of experiments on DC motor position control are performed and control performance of system utilizing SCO link with and without Smith predictor is proved to be higher than control performance of system utilizing ACL link.

Keywords: networked control systems, NCS, bluetooth, SCO

I. INTRODUCTION

NCS (Networked Control Systems) are a feedback control systems where information from the sensors and the controllers is sent over electronic communication network [1], as shown in Fig. 1. Their existence comes from requirement of distance separation between controller, sensors and actuators due to environment scale, hazard conditions, remoteness of expert (remote surgery), etc.

RS-422/485 networks with various protocols like Modbus, Profibus or proprietary ones available from 1970s have been used widely in industry. CAN (Controller Area Networks) utilized by DeviceNet, CANOpen and other protocols have been used widely since 1987. In the last decade Ethernet networks with TCP/IP (Transmission Control Protocol/Internet Protocol) or UDP (User Datagram Protocol) protocols became popular due to spread of Internet. These are mature technologies, but they are wired. However it is not rare for experiment to be quickly performed on-site where all sensors and actuators are placed temporarily and then removed. In that case a mess of wires is undesired and sometimes it even cannot be afforded, as in case of remote-controlled helicopter which obviously has to be wireless controlled.

There are a number of wireless technologies available in market. The most widespread are WiFi (802.11*), cellular (GSM, CDMA, LTE, etc.) and Bluetooth networks due to their public use. ZigBee, raw IEEE 802.15.4 standard based networks and a number of other ones can be found less often. A number of deficiencies can be seen in each technology if applied to NCS. For example, WiFi requires a lot of power,

cellular networks require presence of mobile operator, ZigBee doesn't have low latency by design and compatibility between various vendors [2], etc. Bluetooth has some deficiencies as well, like complicated connection setup, though improved in v3.0 of the standard. While there are so many wireless technologies and all of them have some deficiencies, the Bluetooth seems to be promising technology for wireless control. One of the reasons for that is that the FHSS (Frequency-Hopping Spread Spectrum) technology utilized by Bluetooth is proven to be usable for remote control. The worldwide leader of RC (Remote Controlled) models – Futaba, introduced controllers and receivers utilizing FHSS in 2.4GHz range in 2010. These controllers have been accepted by RC modeling community positively and in this paper the method to achieve high performance of Bluetooth-based controller by utilizing Bluetooth audio channel is discussed.

There are a number of papers utilizing Bluetooth for NCS. But typically experiments in them are performed using SPP (Serial Port Profile) that works on top of ACL (Asynchronous Connectionless) Bluetooth link [3,4]. The main reason for SPP use is implementation simplicity letting parties communicate with each other as they would do being connected by serial cable. But SPP introduces non-linearity into communication process, grouping incoming data bytes in unpredictable way, thus splitting data packets between network packets or combining several data packets into single network packet.

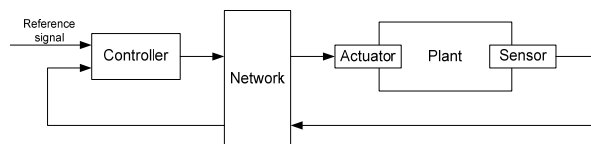


그림 1. 네트워크 제어시스템 블록도.
Fig. 1. A block diagram of an NCS.

* 책임저자(Corresponding Author)

논문접수: 2012. 5. 15., 수정: 2012. 6. 14., 채택확정: 2012. 6. 30.

Ulugbek Umirov, 박정일: 영남대학교 전자공학과
(ulugbek@yu.ac.kr/jipark@yu.ac.kr)

Another deficiency is packet retransmission that leads to growth of latency and its variation. Also there is a drawback of ACL link that if the packet sent from master is corrupted and thus lost, then the packet from slave cannot be sent. Aforesaid leads to Bluetooth-aware control implementation, and to SCO (Synchronous Connection-Oriented) link. Idea of possibility of SCO link utilization was proposed back in 2000 [5], but was passed over.

In [6], Nilsson analyses several important features of NCSs. Nilsson's work standardized a few basic assumptions, which are used throughout this work. First, sensors are always assumed to be time-driven, that is, the plant output is sampled periodically. Second, actuators are event-driven, and therefore apply control signals as soon as they are received. The actuator also holds the last received control signal until a new one is received. Finally, the controller is also event-driven, meaning that the control signal is calculated as soon as a sensor value is received.

This paper consists of several sections, starting with Bluetooth standard description focused on SCO links and latencies. After the Bluetooth description the paper is continued with control problem caused by network latencies and explanation of solution is given in couple with experiments. Then paper is closed by conclusion.

II. BLUETOOTH FOR NCS

Bluetooth [7] is an open wireless technology standard for exchanging data over short distances. Originally it was designed as a wireless alternative to RS-232 data cables in 1994.

1. Bluetooth Standard

Bluetooth piconet consists of single master and up to 7 slaves. Communication is performed only between master and slaves, slaves cannot communicate to each other. Single-slave communication provides lower latency and higher throughput between 2 devices, thus the following focuses on that case. Also the new features of Bluetooth introduced since v1.1 are not covered, since audio channel communication is used and it wasn't changed in new standards.

Bluetooth utilizes TDD (Time Division Duplexing) and is half-duplex system, which means that one node transmits

while the other node receives and vice-versa. Time is divided into slots that are nominally $625\mu\text{s}$ in length (1600 slots per second) and numbered with consecutive integers. The master transmits to the slave in even-numbered time slots, and the slave transmits to the master in odd-numbered time slots. Each transmission takes place at a new hopping frequency (there are 79 frequencies available) and a complete packet of data is sent in each time slot. This process is depicted in Fig. 2.

Each packet is allowed up to $366\mu\text{s}$ for its transmission, equating to a maximum one-slot packet length of 366 bits (payload is 240 bits), due to 1Mbps raw baseband rate. The additional $259\mu\text{s}$ are used by the radio to change to the next frequency in the hop sequence. During the time nominally needed to perform a hop, no communication occurs in the piconet.

Two different physical links can be established between Bluetooth devices: ACL and SCO links.

The SCO link is a circuit-switched, point-to-point link between a master and single slave. Latency is guaranteed to be a small, fixed value through two methods:

- Packets are scheduled for transmission in specific time slots;
- Packets are never retransmitted.

SCO packets are exchanged in pairs, first from master to slave and then from slave to master, in consecutive time slots. The slave can transmit a packet in its reserved slot even if the master doesn't transmit in the previous slot, but cannot if the master transmits a packet to a different slave in that slot.

The ACL link is used where data integrity is more important than latency. Packet switching is used on the ACL link, where a packet received with uncorrectable bit errors is usually retransmitted until it is error free. The average number of retransmissions increases with increasing channel BER (Bit-Error Rate), so latency is variable and can occasionally be quite long.

The analogy between ACL and TCP/IP connection and between SCO link and UDP connection can be built. From papers describing NCS based on Ethernet networks [8] it is known that UDP suits control better than TCP/IP. So does SCO link in Bluetooth piconet.

An SCO link is symmetric between a master and a single slave, so slots are reserved to support a 64kbps data rate in each direction. The payload part of a SCO packet has only one field in it, consisting of 240 bits. Single-slave SCO link utilizes HV1 (High-Quality Voice) packets with (3,1) binary repetition code which essentially transmits each bit 3 times, resulting in 80 available bits for data per packet. Simple calculation of $80 \times 1600 / 2 = 64,000$ (80 data bits per packet multiplied by 1600 Bluetooth time slots and divided by 2 due to half-duplex nature of Bluetooth) leads us to observation that HV1 SCO link fully loads the Bluetooth piconet, occupying all time slots, i.e. we cannot have any other link in presence of such link.

Bluetooth module communicates to host via HCI (Host Controller Interface) transport layer. In this work, we use UART (H4) transport layer, represented by null-modem connection with RTS/CTS (Request to Send/Clear to Send)

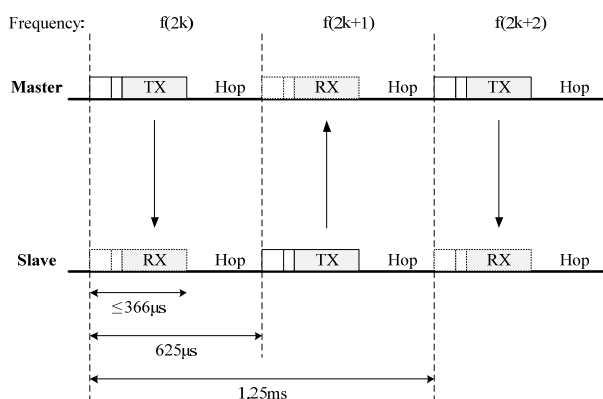


그림 2. Point-to-point 방식에서의 블루투스 TDD통신 과정.

Fig. 2. Bluetooth TDD communication process in a point-to-point configuration.

support.

The baud rate to be used is manufacturer specific, but clearly, it must be high enough to support the speeds at which the host and module must communicate to keep up with the Bluetooth link itself. The serial port should be configured for eight data bits, no parity, one stop bit, and RTS/CTS hardware flow control.

2. Transmission Latency

Transmission latency is composed of various parts, like UART transport latency, time-slot waiting latency, packet transmission latency, processing time, etc. Bluetooth SCO HV1 packet contains 10 bytes of data. Thus the HV1 HCI data packet contains 13 bytes (3 bytes header + 10 bytes data). UART (H4) transport adds 1 byte packet indicator, resulting in 14 bytes per H4 packet. We hit the limits and perform 800Hz control, thus we need to transmit $14 \times 800 = 11,200$ bytes per second through UART. The byte format is 8n1 (8 data bits, no parity bit, 1 stop bit), thus each byte requires 10 bits (1 start bit + 8 data bits + 1 stop bit), resulting in total 112,000bps rate. The minimum standard bit rate larger than 112,000bps is 115,200bps – that is the minimum acceptable UART rate. The latency in network when using this rate can be estimated.

Transmission latency is composed of sender UART transmission latency, waiting for time slot, baseband transmission latency and receiver UART latency.

UART transmission latency is expressed as

$$T_{uart} = \frac{(10 + 3 + 1) \times (1 + 8 + 1)}{115,200} = \frac{140}{115,200} = 1.22ms \quad (1)$$

Waiting for transmission slot can take time between 0 and 1.25ms.

$$0 \leq T_w \leq 1.25ms \quad (2)$$

Baseband transmission will take:

$$T_{bb} = 366\mu s \quad (3)$$

Thus to transmit single HV1 packet from host to host takes

$$\begin{aligned} T &= 2 \times T_{uart} + T_w + T_{bb} \\ &= 2.44ms + [0; 1.25]ms + 0.37ms \\ &= [2.81; 4.06]ms \end{aligned} \quad (4)$$

In the best case it will take 2.81ms to deliver 10 bytes HV1 packet from master to slave and vice versa. In the worst case, when UART packet arrives to module right after the moment its' time slot has started, the time delay is increased by time of 2 Bluetooth bandwidth slots, i.e. by 1.25ms.

The RTL (Round-Trip Latency) consists of 2 transmissions latencies and time for control signal calculation at master side, as depicted in Fig. 3. If fast DSP is used at controller side, then ~10ms RTL can be expected. However, due to inefficiencies of implementation this paper operates with ~20ms RTL.

If there is a requirement to control several slaves, HV2 packets can be used to communicate with 2 slaves; in this case master will exchange packets with first slave, in the next period with second slave, and so on. In this case the achievable control frequency is 400Hz and each packet can

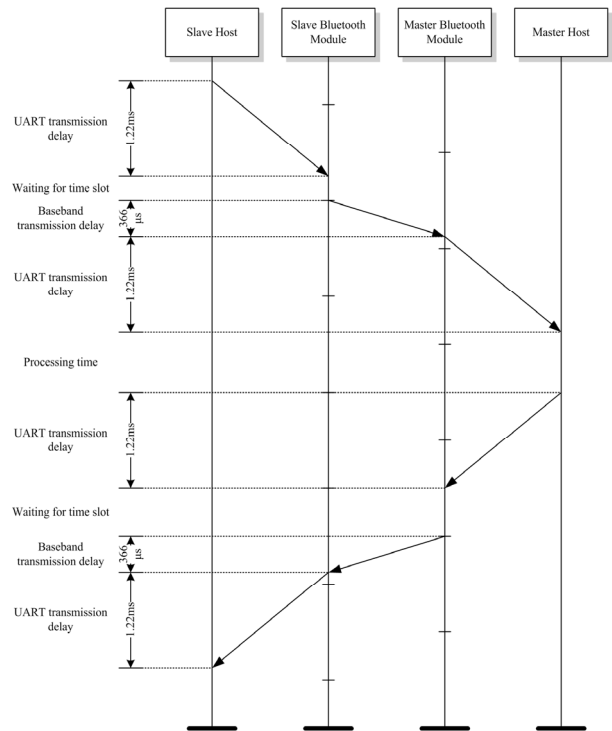


그림 3. 115,200bps UART (H4)를 사용하는 호스트와 통신하는 블루투스 모듈 시스템의 왕복 지연시간.

Fig. 3. Round-trip latency in system with Bluetooth modules communicating to hosts via UART (H4) working at 115,200 bps.

contain up to 160 bits of data protected by shorted Hamming code. Also HV3 packets can be used to communicate with 3 slaves; in this case the achievable control frequency is 266Hz and each packet can contain up to 240 bits of unprotected data, thus the system utilizing such packets has to introduce own corruption protection of data. In both cases the baseband transmission delay remains the same, because all HV packets fit within one time-slot, however the UART transmission delay and waiting for time-slot values are increased, thus the overall RTL is increased.

III. DESIGN OF NCS

1. Setup of NCS

The distributed control configuration is shown in Fig. 1. The feedback loop consists of three parts: the sensor, the controller and the actuator. The sensor node is time-driven while the controller and actuator are event-driven. The sensor samples the process periodically and sends the measurement values to the controller. Upon receipt, the controller calculates a new control signal and sends it to actuator node which outputs the value. In our setup, the sensor node and actuator node are located in the same hardware unit, called the Remote I/O.

The implementation structure of networked control system is shown in Fig. 4. Bluegiga WT11 Bluetooth modules with custom firmware are used and connected to PC and Remote I/O via UART link. Remote I/O is based on AVR ATmega128 MCU with PWM controlled DC motor driver.

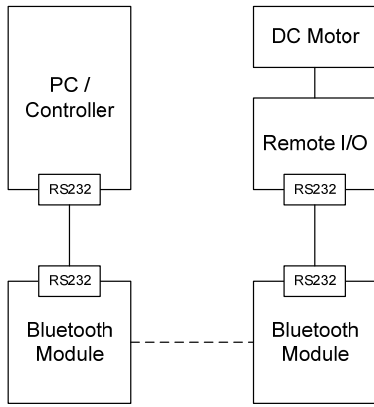


그림 4. 구현된 NCS 구조.
Fig. 4. NCS implementation structure.

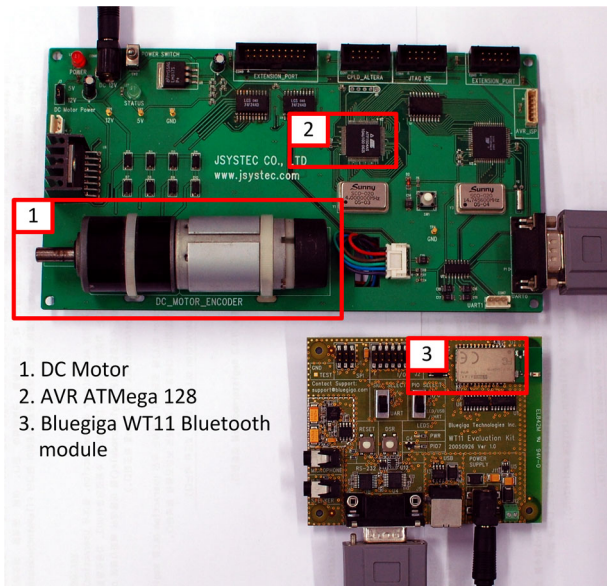


그림 5. 리모트 I/O 시스템과 Bluegiga WT11 블루투스 모듈.
Fig. 5. Remote I/O system and Bluegiga WT11 Bluetooth module.

The photo of Remote I/O system with Bluegiga WT11 Bluetooth module is shown in Fig. 5.

In order to utilize Bluetooth SCO link, the HCI layer has to be utilized or custom firmware setting direct link between the baseband and application layer has to be developed. Bluegiga WT11 module uses CSR (Cambridge Silicon Radio) BlueCore 4 chip and CSR provides BlueLab IDE (Integrated Development Environment) with sample firmware source codes. The sample source code includes firmware code with support for streaming audio data over HCI. However the firmware enables only A-law, μ -law and CVSD (Continuous Variable Slope Delta Modulation) air modes, while Bluetooth standard defines additional transparent data air mode, which has to be added to source code in order to enable setup of SCO-S link (User Synchronous Data Logical Link). After firmware modification the connection setup follows the Bluetooth specification, i.e. it is represented by sequence of the following HCI commands/events: scan/paging for slave, service discovery, ACL link setup (ACL link always has to be established), SCO link setup.

2. DC Motor Controller

The DC motor model is derived during experiment, and has the form of first-order transfer function, where input is torque [N·m], and output is angular velocity [rad/s].

$$G_m(s) = \frac{1450}{s + 150} \tag{5}$$

Encoder is used in the role of sensor, thus the plant has integrator block added to the motor block as shown in Fig. 6.

To cope with loop delay Smith predictor [9] is used, which structure is shown in Fig. 7.

The quick, easy way to understand the controller is to assume that the plant model is perfect, i.e., $G_m(s) = G(s)$, and ignore the feedback from the plant and the feedback from the delayed plant model. Then, the control signal, u , will be the same control signal sent to the plant as if it were controlled in a closed loop without delay. This means that the controller can be designed for the delay-free system, and therefore can have a much more aggressive response. However, because the plant model is not always perfect, the error signal must also include feedback from the plant. The feedback from the plant must be compared to the output of the model, and because the plant has delay, the feedback from the model must also include that delay.

As was mentioned above, single HV1 packet delivers 10 bytes of data. Packets sent from Remote I/O to Controller use 3 bytes out of 10, and have the structure shown in Fig. 8. ID – is sequential number of packet, and Encoder value (16-bit) is split into high and low bytes.

On the other side the packet sent from Controller to Remote I/O contains PWM duty ratio, also 16-bit value split into high and low bytes. There is no ID sent from Remote I/O to Controller. The structure of the packet is shown in Fig. 9.

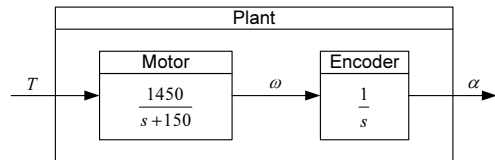


그림 6. 플랜트 모델: T-토크, ω - 각속도, α -출력각도.
Fig. 6. Plant model: T – torque, ω – angular velocity, α – output angle.

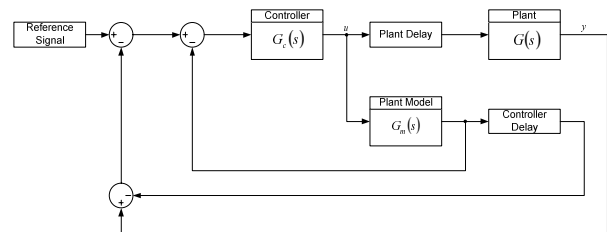


그림 7. Smith 예측기.
Fig. 7. Smith predictor.

ID	Encoder (H)	Encoder (L)	0000000
----	-------------	-------------	---------

그림 8. 원격 I/O에서 제어기로 보내는 패킷 구조.
Fig. 8. Structure of packet sent from Remote I/O to Controller.

PWM duty (H)	PWM duty (L)	00000000
--------------	--------------	----------

그림 9. 제어기에서 원격 I/O로 보내는 패킷 구조.
Fig. 9. Structure of packet sent from Controller to Remote I/O.

3. Experiments

A number of experiments on system have been performed using single DC motor. A conventional discrete PID controller is used for control.

$$G_c(z) = P + I \cdot T_s \frac{1}{z-1} + D \cdot \frac{N}{1 + N \cdot T_s \frac{1}{z-1}} \quad (6)$$

First PID controller is tuned using improved Ziegler-Nichols method [10] for system without latency targeting 2s settling time. Sampling time $T_s=1.25ms$. The obtained PID parameters are $P=0.28$, $I=0.004$, $D=-0.05$ and filter coefficient $N=4.4$.

Then experiments are performed on wired system. Wired system also has RTL due to UART transmission latency and processing time, but its' average is 3ms and is negligible. Experiment result is shown in Fig. 10.

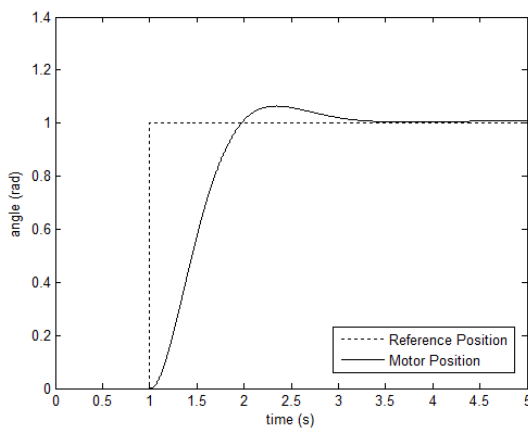


그림 10. 유선 연결을 통한 1개 DC 모터의 PID 제어.
Fig. 10. PID control for one DC motor via wired connection.

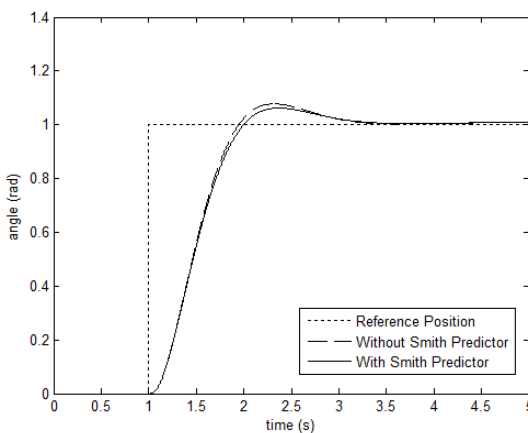


그림 11. 블루투스 SCO 링크를 통한 1개 DC 모터의 PID 제어.
Fig. 11. PID control for one DC motor via Bluetooth SCO link.

For Bluetooth SCO link experiments performed with and without Smith predictor, in order to verify its effect. The Smith predictor is configured for 20ms prediction – same as average RTL of the system. The results of experiments are shown in Fig. 11.

The performance of system with and without Smith predictor is almost identical, due to low latency of SCO link. However system utilizing Smith predictor has smaller overshoot, thus its usage is justified.

Additional experiment has been performed using SPP (Serial Port Profile) of Bluetooth, which is commonly used for Bluetooth based NCSs due to transparency for end-systems, i.e. end-system sends packets like it would send them via wired connection. But SPP uses ACL links and introduces additional layer of logic, improving link reliability on the price of latency increase. Moreover, end-system cannot control packet sending, thus each control/data packet sent by end-systems can be split between Bluetooth baseband packets, introducing additional variation of latency. It was revealed that RTL of Bluetooth system using SPP profile was in [65;95]ms range. Smith predictor's prediction time is set to 80ms. Results of this experiment are shown in Fig. 12.

As can be noticed in Fig. 12, due to increased RTL and its' variation, the overshoot and settling time of the system is increased.

RMSE (Root Mean Squared Error) values are calculated for above experiments and shown in Table 1. It can be observed that system utilizing Bluetooth SCO link has performance close to system utilizing serial wired connection. Lower RMSE values of system without Smith predictor are explained by more aggressive response of system to step function.

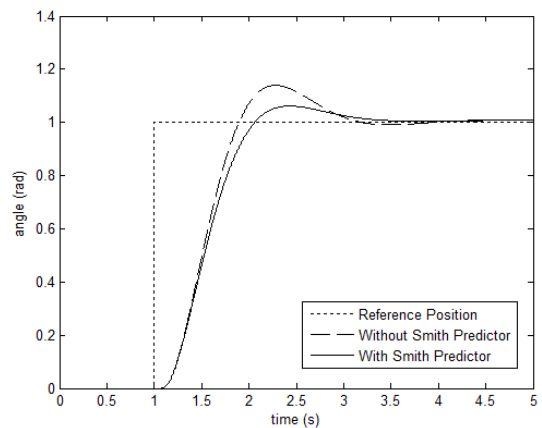


그림 12. 블루투스 ACL 링크(SPP)를 사용한 1개 DC 모터의 PID 제어.

Fig. 12. PID control for one DC motor via Bluetooth ACL link (SPP).

표 1. 1개의 DC 모터 제어시의 루트 평균 제곱 위치 오차.

Table 1. Root mean squared position error for one DC motor control.

	Without Smith predictor	With Smith predictor
Wired	0.2595 rad	-
SCO link	0.2648 rad	0.2662 rad
ACL link (SPP)	0.2850 rad	0.2876 rad

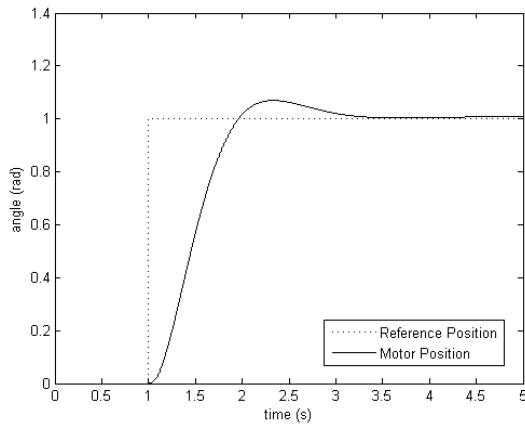


그림 13. 유선 연결을 통한 3개 DC 모터의 PID 제어.

Fig. 13. PID control for 3 DC motors via wired connection.

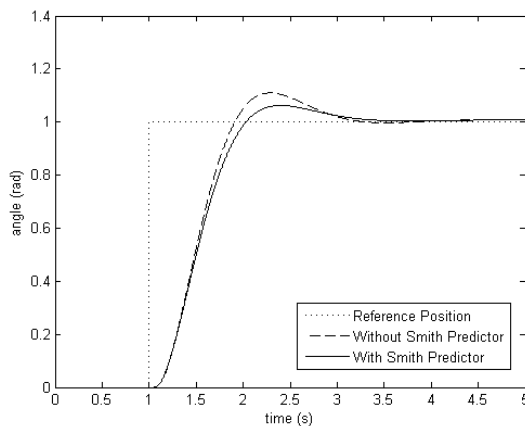


그림 14. 블루투스 SCO 링크를 통한 3개 DC 모터의 PID 제어.

Fig. 14. PID control for 3 DC motors via Bluetooth SCO links.

표 2. 3개 DC 모터 제어시의 루트 평균 제곱 위치 오차.

Table 2. Root mean squared position error for 3 DC motor control.

	Without Smith predictor	With Smith predictor
Wired	0.2617 rad	-
SCO link	0.2759 rad	0.2785 rad

Experiment with 3 slaves utilizing HV3 packets has been performed. Packets content has been tripled, resulting in (3,1) binary repetition code for corruption protection. Sampling time is tripled as well and results in $T_s=3.75\text{ms}$. The PID controller is tuned again targeting 2s settling time. The obtained PID parameters are $P=0.24$, $I=0.004$, $D=-0.05$ and filter coefficient $N=4.4$, it can be noticed that only P parameter is changed.

The wired control system has 9ms average RTL. Results of experiments on wired system are given in Fig. 13 and Table 2. Fig. 13 is almost identical to Fig. 10 due to negligible difference between parameters of single-slave and triple-slave systems.

The Smith predictor for the system is configured for 53ms prediction – the average measured RTL. The results are provided in Fig. 14 and Table 2.

The system with 1 SCO link and system with 3 SCO links

outperform the system with 1 ACL link in terms of position overshoot and RMSE.

IV. CONCLUSION

NCSs, their advantages and deficiencies are discussed. Bluetooth ACL and SCO links are described and latency of SCO link is estimated. The SCO link has low latency with small variation due to fixed packet scheduling is verified via experiments. On top of SCO link more simple control algorithms can be used, which can be effectively tuned by engineers. As a proof of aforesaid a number of experiments using PID control of DC motor position were performed. It is shown that performance of wireless control on top of SCO link is close to performance of wired control.

The main disadvantage of SCO link is that typically Bluetooth modules support SCO link only for audio data, while for control purposes the SCO-S link for user data is required, thus it may be required to modify Bluetooth module firmware. Also setup of SCO link involves preceding setup of ACL link, thus the communication establishment procedure of SCO link is more complex and involves good understanding of control flow of Bluetooth. However, achievable 800Hz low-latency control outweighs implementation difficulties.

Packet-drop impact on control system performance was not covered in this paper as it was negligible in given circumstances. However there are situations possible when packet-drop rate is significant and it is the topic for further research.

REFERENCES

- [1] D. Hristu-Varsakelis and W. S. Levine, *Handbook of Networked and Embedded Control Systems*, Springer Verlag, 2005.
- [2] P. Ling, "Mixed signals - ZigBee aims to be a standard for industrial low-power radio, but we might never see full compatibility between modules that support it," *Electronics Systems and Software*, vol. 5, no. 1, pp. 34-37, Feb.-Mar. 2007.
- [3] K. H. Seol and Y. S. Lim, "Implementation of novel application using Bluetooth," *Proc. of the SICE 2004 Annual Conference*, vol. 2, pp. 1617-1620, 4-6 Aug. 2004.
- [4] F. Mitome and K. Ohnishi, "Bilateral control with bluetooth and its evaluation," *Proc. of the 4th International Conference on HSI (Human System Interactions)*, pp. 346-351, 19-21 May 2011.
- [5] U. Bilstrup and P.-A. Wiberg, "Bluetooth in industrial environment," *Proc. of the IEEE International Workshop on Communication Systems*, pp. 239-246, 2000.
- [6] J. Nilsson, "Real-time control systems with delays," *Ph.D. thesis*, Department of Automatic Control. Lund Institute of Technology, 1998.
- [7] Bluetooth SIG Specification of the Bluetooth System v1.1, 2001.
- [8] L. Cheng and Z. Wang, "Research on delay of data streams in internet-based NCS," *Proc. of the 8th World Congress on Intelligent Control and Automation (WCICA)*, pp. 1846-1851, 7-9 July 2010.
- [9] B. Kristiansson, "Robust PI and PID controllers including Smith predictor structure," *Proc. of the American Control Conference*, vol. 3, pp. 2197-2002, 2001.
- [10] C. C. Hang, "Refinements of the Ziegler-Nichols tuning formula," *Control Theory and Applications*, IEE Proceedings D, vol. 138, no. 2, pp. 111-118, Mar. 1991.



Ulugbek Umirov

2003년 우즈베키스탄, 타슈켄트 정보기술대학(TUIT) 정보기술학과 졸업. 2006년 영남대학교 대학원 전자공학과(공학석사). 2006년~현재 영남대학교 대학원 전자공학과 박사과정. 관심분야는 무선 네트워크 제어 시스템, 기계학습.



박정일

1981년 경북대학교 전자공학과(공학사). 1983년 서울대학교 대학원 전자공학과(공학석사). 1989년 서울대학교 대학원 전자공학과(공학박사). 1989년~1992년 삼성종합기술원 선임연구원. 1992년~현재 영남대학교 전자공학과 교수. 관심분야는 지능제어, 정밀모션제어, 네트워크 제어 시스템.