

논문 2020-15-13

지그비 기반의 실시간 무선 네트워크 모터 제어시스템 (ZigBee-based Real-time Wireless Networked Motor Control System)

박 정 일*
(Jung-Il Park)

Abstract : This paper finds solutions for using ZigBee in wireless networked control system (WNCS). The round trip time delay and packet loss rate of the WNCS are measured. On the basis of these measured data, a playback buffer is used to solve the variable time delay in WNCS, and a Smith predictor is introduced to compensate for the time delay. The WNCS was able to be actually constructed to perform DC motor position control with 40 Hz sampling frequency.

Keywords : NCS, WNCS, ZigBee, Playback buffer, Smith predictor

1. 서 론

네트워크 제어 시스템 (Networked Control System : NCS)은 그림 1에 나타난 것처럼 네트워크를 통하여 피드백 제어를 수행하는 시스템을 말한다 [1]. 최근에는 블루투스 (Bluetooth)나 지그비 (ZigBee) 등의 근거리 무선 네트워크로 페루프 제어 시스템을 구성하면 무선 네트워크 제어 시스템 (Wireless Networked Control System : WNCS)이라고 한다 [2-7].

N. Boughanmi는 WNCS에서 지그비를 사용하기 위해서 CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance), Black burst, 비컨 사용 GTS (Guaranteed Time Slot) 메커니즘의 세 가지 솔루션을 제시하였다 [2]. Y. Millan은 NCS의 매체 종류와 특히 WNCS에 사용가능한 지그비, 블루투스 프로토콜과 이들 매체에서 발생하는 시간지연 등을 모델링하고 Simulink로 시뮬레이션 하였다 [3]. J. Park은 블루투스와 지그비 두 가지의 무선매체에 대해서 WNCS를 구현하기 위한 여러 가지 솔루션을 제시하였다 [4]. P. Park은 지그비 네트워크에서 일어나는 시간지연, 패킷 손실 확

률을 성능지수로 잡고 최적 제어를 찾아내는 방법을 제시하였다 [5]. B. Park은 WNCS에서 고려되어야 할 클러스터링, 스케줄링 등을 총체적으로 고려한 견실 무선 센서 액추에이터 네트워크 (R-WSAN: Robust Wireless Sensor and Actuator Network)를 제안하였다 [6]. 이들 WNCS에 관한 선행 연구들은 Simulink의 TrueTime 블록을 사용한 시뮬레이션 [2, 3]이거나, 가변 시간지연과 패킷 손실 등을 고려한 모델을 이용하여 제어를 설계하고 시뮬레이션 하였다 [5, 6]. E.P. Godoy는 CAN과 지그비를 사용한 WNCS의 실제 테스트 베드를 만들기도 했다. 하지만 구현된 시스템의 샘플링 주파수는 고작 10Hz (100msec)에 불과하였다 [7].

WNCS는 센서로부터의 데이터 량이 많지 않고, 제어패킷을 자주 보내야 하는 특징이 있어서, 패킷 전송률을 높이고 지연시간을 줄이는 것이 중요하다. 따라서 본 논문에서는 시간지연을 줄이는 솔루션을 찾는데 주안점을 두었다. 또 네트워크의 대역폭, 데이터를 전송할 노드의 수가 여러 개일 때 우선순위 결정으로 인한 전송에서의 지연시간이 불규칙하게 발생한다. 본 논문에서는 이러한 시간지연과 불확정성을 직접 측정하고, 이들을 근거로 Smith 예측기 (Smith predictor) [8]와 재생버퍼 (Playback buffer) [9]를 도입한다. 실제로 DC 모터의 위치제어를 수행하는 WNCS를 구축하여 설계한 제어기의 성능을 검증한다.

*Corresponding Author (jipark@yu.ac.kr)

Received: Jan. 22, 2020, Revised: Mar. 23, 2020,

Accepted: Mar. 26, 2020.

J.I. Park: Yeungnam University (Prof.)

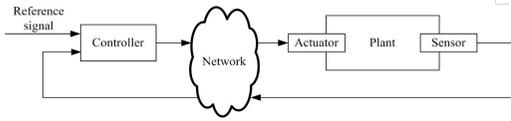


그림 1. 네트워크 제어시스템 블록도
Fig. 1 Block diagram of an NCS

II. WNCs에서의 지그비 적용

지그비는 WPAN (Wireless Personal Area Networks)용 IEEE 802.15.4 표준을 기반으로 하는 소형, 저전력 고급 통신 프로토콜로 저속 애플리케이션용으로 설계되었다. 2.4GHz 무선 대역의 채널 대역폭은 250kbps이다 [10].

IEEE 802.15.4는 여러 장치가 동일한 주파수 채널을 사용하기 위해서 경합기반과 경합이 없는 두 가지 채널 액세스 메커니즘을 사용한다. 경합기반 방식에서는 CSMA-CA방식을 사용한다. 이 방식은 장치가 전송하기를 원할 때마다 먼저 다른 장치에서 채널을 사용하지 못하도록 CCA (Clear Channel Assessment)를 수행한 후 자체신호 전송을 시작한다. 경합이 없는 방식에서의 코디네이터는 특정 슬롯을 특정장치에 할당한다. 이를 Guaranteed Time Slot (GTS)이라고 하며, 할당된 GTS를 갖는 장치는 CSMA-CA 메커니즘을 사용하지 않고 GTS 동안 전송을 한다. GTS를 제공하기 위해서는 코디네이터는 비컨 메시지를 전송하여 네트워크의 모든 장치를 동기화한다. 이러한 네트워크를 비컨 사용 PAN이라고 한다.

NCS에서는 패킷이 양방향으로 연속적으로 전송되어야 하므로 두 가지 옵션을 설정할 수 있다. 비컨을 사용하지 않는 네트워크로 설정해서, 디바이스가 제어패킷을 받기 위해 지속적으로 코디네이터를 폴링하거나, 비컨 사용 네트워크로 설정해서 코디네이터가 디바이스로 데이터 요청을 트리거하기 위해 지속적으로 비컨을 보내는 방식을 사용할 수 있다. 본 논문에서 사용하는 지그비 XBee 모듈의 GTS 정보가 제대로 제공되지 않아 비컨을 사용하지 않는 모드로 실험을 수행한다. 그러나 데이터 요청에 대한 승인은 계속 전송되어야 하고, 모든 패킷을 보내기 전에 CSMA-CA 절차가 수행된다.

지그비는 스타, 클러스터 트리 및 메쉬 네트워크를 지원한다. 클러스터 트리 및 메쉬 네트워크의 각 홉(hop) 재전송이 패킷지연을 증가시켜서 제어 성능에 좋지 않은 영향을 미치기 때문에 1개의 스타

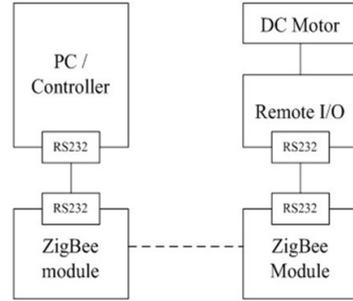


그림 2. 구현된 WNCs 구조
Fig. 2 Implemented WNCs structure

네트워크만으로 여러 슬레이브를 제어하는 방식이 바람직하다. 프로세스에서 패킷은 계속해서 전송되므로 패킷손실이 생기더라도 무방하다. 또 재전송은 추가지연을 발생시키므로 ACK 메커니즘을 사용하지 못하게 한다. 이처럼 ACK를 비활성화 시키면 지연을 최소화할 뿐만 아니라 채널 처리량을 절약할 수 있다. 그러나 일부 공급업체의 지그비 모듈은 유니캐스트 패킷에 대한 ACK 비활성화를 허용하지 않으므로 의도적으로 ACK를 발생시키지 않는 브로드캐스트 패킷을 사용할 수도 있다. 지그비는 저전력 프로토콜이지만, WNCs의 지연 최소화를 위해서 절전모드를 해제하고 사용하는 것이 바람직하다.

III. WNCs 설계 및 실험

WNCs는 제어가 들어있는 상위 PC제어기와, 센서와 액츄에이터를 포함하는 타겟보드인 원격 I/O장치로 구성된다. 본 실험에서는 정확한 클럭이 동작하고 있는 타겟보드의 클럭을 분주하여 샘플링 타임을 발생하였으며, 이 샘플링 시각마다 센서노드는 측정된 센싱 데이터를 상위 PC제어기로 전송하고 (Time-driven 방식), 상위 PC제어기에서는 이 센싱 데이터가 전송되어 들어올 때 마다 제어입력을 계산해서 바로 타겟보드로 전송하면, 액츄에이터는 이를 받는 대로 동작한다 (Event-driven 방식). 구현된 WNCs의 구조를 그림 2에 나타낸다.

지그비 MaxStream XBee/XBee2는 115kbps로 RS-232 링크를 통해 PC제어기와 원격 I/O장치에 연결된다. API (Application Programming Interface) 프레임이 모듈의 구성과 통신을 제어하는데 사용된다. 원격 I/O장치는 PWM 모터 드라이버와 AVR ATmega128 MCU로 구성되어 있다. 카운터와 모든 주변회로들은 FPGA모듈 내에 집적하

ID	Packet #	Encoder High	Encoder Low	Real Time
----	----------	--------------	-------------	-----------

ID	Packet #	PWM High	PWM Low
----	----------	----------	---------

그림 3. 패킷구조

(위) 원격 I/O→제어기, (아래) 제어기→원격I/O

Fig. 3 Packet format (up) Remote

I/O→Controller , (down) Controller→Remote I/O

였다. 상위 제어기에서 원격 I/O장치로 보내는 패킷은 ID, 패킷 번호, PWM 상위/하위 값을 나타내는 4개의 바이트로 이루어져 있으며, 원격 I/O장치에서 상위 제어기로 보내는 패킷은 ID, 패킷 번호, 모터 엔코더의 카운트 값 상위/하위 바이트, 실시간을 나타내는 5개 바이트로 되어 있다. 두 경우 모두 패킷 사이즈를 줄이기 위해서 스타트/스톱 바이트를 사용하지 않았다. 그 패킷 구조는 그림 3에 나타내었다.

1. WNCS의 왕복지연시간 및 패킷 loss 측정

원격 I/O장치에서 상위제어기로 전송하는 패킷속에 실시간 정보를 저장하여 보냄으로써, 구동신호가 도착한 시점에서의 시간과 패킷에 실려 있는 시계열 인덱스의 시간차를 계산함으로써 왕복 지연시간 (RTT : round-trip time)을 구할 수 있다.

패킷 전송 오류 발생시 재시도를 하지 않는 것이 실시간 제어에 유리하므로, 지그비 모듈의 API (Application Programming Interface) 모드를 사용하였다. 이렇게 하면 1 개의 패킷으로 전송하는 것이 가능하고, 더구나 API 모드는 1 개 노드와 복수개의 노드와 통신이 가능하도록 해준다. 또한 패킷 사이즈를 줄이기 위해서 ASCII 형식에서 바이너리 형식으로 바꾸어 그림 3에서 나타낸 것처럼 제어패킷은 4바이트, 정보패킷은 5바이트를 사용하였다. 패킷 전송을 위해서 2번의 CCA 재시도와 3번의 ACK 재시도를 필요로 한다. 그러나 이는 실시간 전송에는 역효과적이다. MaxStream XBee Series 1 모듈은 유니캐스트 모드에서는 CCA와 ACK 재전송을 없애는 것이 불가능하다. 그러나 Series 2 모듈은 브로드캐스트 모드를 사용할 수 있으며 이 경우에는 ACK 재시도를 하지 않는다. 하지만 이 모드의 경우 매번 패킷을 두 번씩 전송하여 많은 대역폭을 소모한다. 본 실험에서는 Freescale 칩을 기반으로 한 MaxStream XBee Series 1 모듈을 사용하여 MAC 층의 파라미터를 수정하면 CCA 재

표 1. 지그비의 왕복지연시간 측정
Table 1. RTT measurement of ZigBee

Sample period (ms)	# of nodes	Average RTT(ms)	Max. RTT (ms)
25	1	41.74	48
	2	>255	>255
	3	N/A	N/A
50	1	40.27	49
	2	46.97	60
	3	128.79	>255
75	1	40.23	48
	2	49.42	59
	3	49.96	84
100	1	40.31	48
	2	42.36	48
	3	46.38	54
200	3	41.00	49

시도와 ACK 기능을 없앨 수 있었다. 모든 모듈은 지그비 채널 #16 에서 동작하고 네트워크 내에서 코디네이터가 없다. 즉, 비컨을 사용하지 않는 네트워크가 가능하다. 값이 255 (1 바이트) ms 보다 크면 포화가 되고, 0은 시간지연 계산이 불가능함을 나타낸다. 여러 샘플링 주기에 대해서 또 원격 I/O 노드의 수를 변경시켜가면서 왕복지연시간을 측정 한 결과를 표1 에 나타내었다.

표 1에서 알 수 있듯이 샘플링 주기가 25ms인 경우에는 단지 1개의 노드만을 제어할 수 있었으며, 2개의 이상의 노드인 경우에는 RTT가 너무 커서 측정이 불가능하였다. 샘플링 주기가 50ms인 경우에는 2개 노드까지 제어하는 것이 가능하지만 3개의 노드인 경우에는 제어가 불가능하였다. 75ms 의 샘플링 주기인 경우에는 3개의 노드까지 제어가 가능하였지만 때로는 지그비 모듈의 버퍼의 오버플로우가 발생하여 리셋이 될 때까지 동작을 멈추는 현상이 발생하였다. 또한 3개의 노드에 대해서 안정한 제어를 위한 통신을 위해서는 샘플링 주기가 100ms가 되어야 했다. 즉, 10Hz 의 주기로 제어를 하는 것인데 이는 DC 모터를 실시간으로 제어하기에는 충분한 제어 샘플링 주기가 못된다. 이러한 전체적인 결과로 미루어 네트워크가 부하가 별로 없다면 전형적인 RTT는 약 40ms 정도였다. 이러한 왕복 지연시간의 분포를 그림 4에 나타내었다.

1 개의 원격 I/O를 사용하는 경우에 CCS 재시도와 ACK 기능을 사용하지 않는다면 데이터 전송률을 채널허용치 이하로 유지할 수 있어서 패킷 손실의 가능성은 낮아진다. 그러나 여러 개의 원격

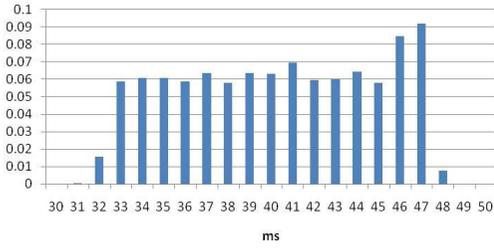


그림 4. 패킷 전송 오류시 재시도를 하지 않는 경우의 RTT 분포

Fig. 4 RTT distribution without retry in case of packet dropout

I/O를 사용하는 경우에는 CCA 재시도를 하지 않기 때문에 패킷 손실의 가능성이 높아진다. 어느 한 모듈이 패킷을 전송하려고 할 때 다른 모듈이 전송도중임을 발견하면 자신의 패킷을 버린다. 또한 패킷 손실은 turn-on 시간에도 관계한다. 만약 같은 샘플 시각에 두 개의 time-driven 시스템이 동시에 turn-on 된다면 한 모듈은 패킷을 전송할 것이고 다른 모듈은 패킷을 drop하기 때문에 패킷 손실의 확률은 더 높아진다. 실시간 제어상태에서는 데이터가 빨리 과거의 정보가 되기 때문에 CCA 재시도를 할 필요가 없으므로 CCA 재시도 기능을 disable 시킨다. RTT 측정 실험을 수행하면서 동시에 패킷 손실을 측정하여 그 결과를 표 2에 나타내었다. 결과를 요약하면, 지그비의 최대 데이터 전송률은 ACK없이 최대 허용 페이로드를 가진 패킷의 단일 방향 전송에서 80kbps이다. 이것도 1 개의 패킷으로 최대 허용된 데이터 길이 (100바이트)를 사용할 때만 얻을 수 있다. 양방향으로의 데이터 전송률은 20kbps 정도이다. 본 논문에서의 WNCS의 실시간 제어에서는 4~5바이트의 패킷을 보내며, ACK가 없는 단일방향 전송에서는 패킷을 5ms마다 전송시켜 8kbps (200×5×8=8,000 bit/sec)를 얻을 수 있었다. WNCS는 양방향 전송이므로, 두 노드 사이에 5바이트와 4바이트를 번갈아 보내는 실험에서는 25ms (40Hz)마다 보낼 수 있어서 2,880bps (40×9×8)의 전송률을 얻을 수 있었다. 더구나 복수개의 노드를 갖는 경우에 제어를 하면 제어 샘플링 주파수가 더 낮아지게 된다. 3개의 노드인 경우는 10Hz정도에 불과하였다.

2. WNCS 모터제어 실험

정확한 플랜트 모델을 얻을 수 있다면 시간지연이 루프 내에 있을 때, 특히 시간지연이 길 때는

표 2. 지그비의 패킷 손실 측정

Table 2. Packet loss measurement of ZigBee

Sample period (ms)	# of nodes	Minimum Packet loss (%)	Maximum Packet loss (%)
25	1	0.00	0.15
	2	64.19	75.33
	3	N/A	N/A
50	1	0.00	0.15
	2	9.36	33.40
	3	17.12	38.53
75	1	0.00	0.15
	2	0.10	24.57
	3	18.84	32.92
100	1	0.00	0.15
	2	0.10	19.41
	3	2.07	31.70
200	3	0.20	16.67

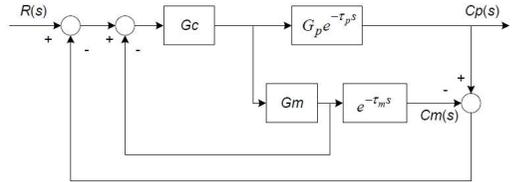


그림 5. Smith 예측기

Fig. 5 Smith predictor

Smith 예측기를 사용하여 페루프 시스템의 성능을 획기적으로 개선시킬 수 있다. 이는 루프 내에 지연이 없는 것처럼 시스템을 제어하기 위한 플랜트 모델과 제어기 지연을 사용한다 [8]. 그림 5는 Smith 예측기의 구조를 나타낸다.

플랜트 모델이 플랜트와 같으면 (Gp=Gm) 플랜트로부터 직접 피드백을 시키지 않고 플랜트 모델로부터 피드백을 시킨다. 제어 입력신호는 마치 페루프에 지연이 없는 것처럼 플랜트에 인가된다. 제어기는 지연과는 무관하게 설계되어서 훨씬 우수한 응답을 얻을 수 있게 된다. WNCS는 루프 내에 시변·랜덤인 지연을 포함한다. 플랜트 루프내의 지연과 제어기 속의 지연모델이 차이가 성능을 나쁘게 하고 시스템을 불안정하게 만든다. 시변인 시간지연을 극복하기 위해서 액츄에이터로 제어신호가 일정한 시각마다 인가되도록 재생 버퍼 (Playback buffer)를 이용하였다. 재생버퍼는 원래 무선 미디어의 스트리밍에서 지연과 지터를 해결하기 위해서 사용된 것이다. 이를 WNCS에 도입하면 루프 내의

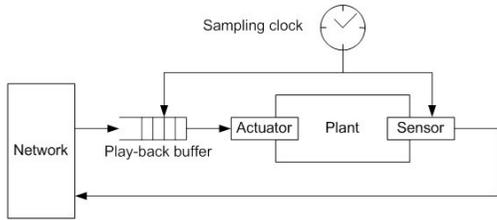


그림 6. Playback 버퍼를 갖는 원격 I/O
Fig. 6 Remote I/O with playback buffer

지연시간이 일정하게 되어 훨씬 확정적으로 된다. 그러나 지연시간이 더 길어지게 되어 불안정하게 만드는 요인이 될 수도 있다. V. Liberatore는 제어 알고리즘과 액츄에이터, 센서를 위한 네트워크 제어에 재생버퍼를 사용하여 그 효과를 시뮬레이션을 통하여 검증하였다[9]. 주요 특징은 그림 6에 나타난 것처럼, 구동신호를 설정한 재생시점에 도달할 때까지 기다렸다가 액츄에이터로 인가하는 것이다. 그러면 루프내의 시간 지연 변화가 제거되어 예측 제어를 설계하기가 쉬워지게 된다. 본 논문에서는 RTT가 30~50ms로 분포하므로 재생버퍼의 재생시간을 50ms로 설정하여 DC모터의 위치제어 실험에 적용하였다.

DC 모터의 위치제어 실험에서 왕복 시간 지연은 제어 노드의 수에 따라서 달라지므로 여러 시간으로 하였으며 이 시간 내에 액츄에이터에 도달하지 않은 구동 신호는 버리고 전 샘플링 타임의 구동 신호를 그대로 인가하였다. DC 모터의 모델은 실험으로 구하였으며 (1) 식과 같은 1차 전달함수를 얻었다. 여기서 입력은 토크 [N.m]이고 출력은 각속도 [rad/s]이다.

$$G_m(s) = \frac{1450}{s+150} \quad (1)$$

DC Motor는 dead-zone을 감소시키기 위해서 높은 기어비를 사용하고 있어서 기준신호 $\pi/2$ rad의 다양한 프로파일에 대해서 위치제어 실험을 하였다. (2) 식으로 표시되는 PI 제어를 사용하였다.

$$G_c(z) = K_p + K_i \cdot T_s \frac{1}{z-1} \quad (2)$$

PI, PI+ SP, PI+ SP+ PB (PI : PI controller, SP : Smith predictor, PB : Playback buffer) 3 가지 경우로 실험을 수행했다. 모든 제어에 정상상태 오차를 줄이기 위해 작은 적분이득을 사용했으며, 이

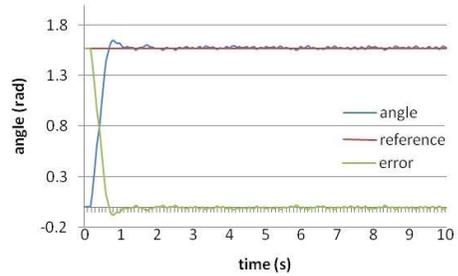
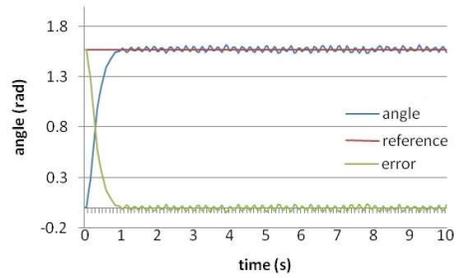


그림 7. PI+ SP (위)와 PI+ SP+ PB (아래) 제어기로 재생버퍼 효과 비교 (Ts=50ms, PB=SP=150ms, Kp=20, Ki=0.1)

Fig. 7 Effect comparison of playback buffer using PI+ SP (up) and PI+ SP+ PB (down) controller (Ts=50ms, PB=SP=150ms, Kp=20, Ki=0.1)

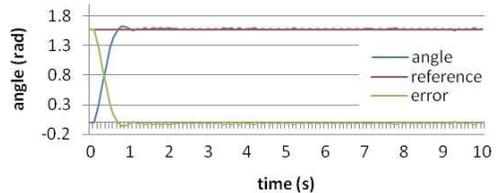
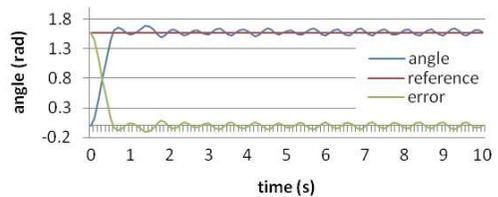


그림 8. 스텝입력에 대한 3개 원격 I/O의 PI (위)와 PI+ SP+ PB (아래) 응답 (Ts=100ms, PB=SP=100ms, Kp=10, Ki=0.1)

Fig. 8 PI (up) and PI+ SP+ PB (down) responses of 3 remote I/Os for step input (Ts=100ms, PB=SP=100ms, Kp=10, Ki=0.1)

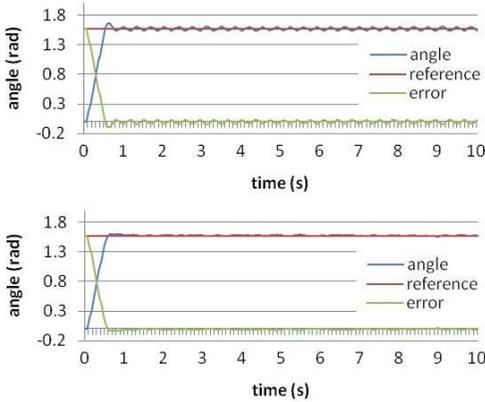


그림 9. 스텝입력에 대한 1개 원격 I/O의 PI (위)와 PI+ SP+PB (아래) 응답 ($T_s=25ms$, $PB=SP=50ms$, $K_p=10$, $K_i=0.1$)

Fig. 9 PI (up) and PI+ SP+PB (down) responses of single remote I/O for step input ($T_s=25ms$, $PB=SP=50ms$, $K_p=10$, $K_i=0.1$)

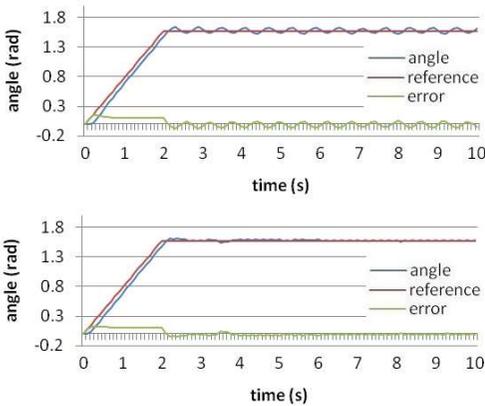


그림 10. 사다리꼴 입력에 대한 3개 원격 I/O의 PI (위)와 PI+ SP+PB (아래) 응답 ($T_s=100ms$, $PB=SP=100ms$, $K_p=10$, $K_i=0.1$)

Fig. 10 PI (up) and PI+ SP+PB (down) responses of 3 remote I/Os for trapezoidal profile ($T_s=100ms$, $PB=SP=100ms$, $K_p=10$, $K_i=0.1$)

로 인한 큰 오버슈트를 방지하기 위해 누적된 오차의 값은 정해진 값 이내로 제한하였다. 3개의 모터를 동시에 제어할 경우에는 샘플링 타임 (T_s)을 100ms로 하였으며, 1개만의 모터를 제어할 경우에는 샘플링 타임을 25ms로 하였다.

그림 7을 살펴보면 재생버퍼를 사용했을 경우가 예상대로 결과가 더 좋았다. 따라서 이후 모든 비교실험에서는 재생버퍼를 사용해서 실험을 수행하였다. 그림 8과 그림 9를 살펴보면, 그림 9의 경우가 더 우수함을 알 수 있다. 그림 9의 경우는 1개의 원격 I/O를 제어하기 때문에 샘플링 시간을 25ms로 줄일 수 있었기 때문이다. 입력 프로파일을 바꾸어 사다리꼴에 대해서 실험을 수행한 결과를 그림 10에 나타내었다. 그림 10의 경우, Smith 예측기와 재생버퍼를 사용한 효과를 살펴보면, 과도상태에서는 실효오차가 0.116 [rad]으로 같았으나 정상상태에서는 0.024 [rad]에서 0.003 [rad]으로 1/8로 줄어들었다.

IV. 결론

본 논문에서는 지그비를 WNCs의 실시간 제어에 사용하기 위한 솔루션을 찾았다. 시판되고 있는 지그비 모듈의 GTS 정보가 제공되지 않아 비컨을 사용하지 않는 모드를 사용하였으며, 또 CCA제시도와 ACK기능을 사용하지 않도록 설정하였다. 이 상태에서 WNCs의 왕복지연시간의 특성과 패킷손실률을 측정해 본 결과, 1개 노드를 사용했을 경우, 왕복지연시간은 30~50ms, 패킷 손실은 0.15%였다. 하지만 제어할 노드가 늘어날수록 왕복지연시간과 패킷 손실률이 높아졌다. 이 값을 근거로 재생버퍼와 Smith 예측기를 사용하여, DC모터의 위치제어를 실시간으로 수행하여 그 성능을 검증하였다. 4바이트의 제어패킷, 5바이트의 정보패킷을 사용하여 1개의 노드만을 제어할 경우에는 40Hz, 3개의 노드를 제어할 경우에는 10Hz의 샘플링 주파수로 DC모터의 위치제어를 수행 할 수 있었다.

References

- [1] S. Zampieri, "Trends in Networked Control System," Proceedings of the International Federation of Automatic Control World Congress, Vol. 41, No. 2, pp. 2886-2894, 2008.
- [2] N. Boughanmi, Y. Song, E. Rondeau, "Wireless Networked Control System Using ZigBee/IEEE 802.15.4," Proceedings of International Federation of Automatic Control, Vol. 42, No. 2, pp. 313-318, 2009.
- [3] Y. A. Millan, F. Vargas, F. Molano, E.

- Mojica, "A Wireless Networked Control Systems Review," Proceeding of IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, pp. 1-6, 2011.
- [4] J. Park, "Applicability of Bluetooth and ZigBee in Wireless Networked Control System, IEMEK J. Embed. Sys. Appl., Vol. 13, No. 1, pp. 9-15, 2018.
- [5] P. Park, J. Araujo, K.H. Johansson, "Wireless Networked Control System Co-Design," Proceeding of IEEE International Conference on Networking, Sensing and Control, pp. 486-491, 2011.
- [6] J. Nah, J. Choi, I. Yoon, P. Park, "Robust Wireless Sensor and Actuator Networks for Networked Control Systems," Journal of Sensors, Vol. 19, No. 7, pp. 1-28, 2019.
- [7] E.P. Godoy, T.A. Oliveria, I.S. Diniz, "A Testbed for Wireless Networked Control Systems Based on CAN and ZigBee," Proceeding of 22nd International Congress of Mechanical Engineering, pp. 3-7, 2013.
- [8] O.J. Smith, "Closer Control of Loops with Dead Time," Journal of Chemical Engineering Progress, Vol. 53, No. 5, pp. 217-219, 1959.
- [9] V. Liberatore, "Integrated Play-Back, Sensing, and Networked Control," Proceeding of IEEE Conference on Computer Communications, pp. 1-12, 2006.
- [10] ZigBee Alliance, ZigBee 2007 Specification, 2007.

Jung-II Park (박 정 일)



He received the B.S. degree in Department of Electronic Engineering from Kyungpook National University in 1981, and the M.S. and Ph.D. degrees in Department of Electronic Engineering from Seoul National University in 1983 and 1989, respectively. He is currently professor in Department of Electronic Engineering at Yeungnam university. His primary research interests include intelligent control, high precision motion control, and wireless networked control system.

Email: jipark@yu.ac.kr