

図 9.17 被験者 3 2 回目

#### 9.8.4 考察

歩行支援機のハンドルに被験者が加える力は、多少のばらつきはあるもののだいたい一定の値をとっており、大きな変化はしていない。大きく力が変化しているところは、 $F_x/F_z$  の値がなんらかの原因で下がったために、速度が遅くなったので力を加えて元の速度に戻そうとしたのではないかと考えられる。データが大きく変化した原因としては、ハンドルに加わる力を検出するセンサに感圧センサを使用したため、ハンドルを持たないときには感知しないようハンドルとセンサの間にすきまを作ったので、ハンドルがしっかりと固定されずずれてしまい、その結果センサがしっかりと力の検出ができなかったのではないかと考えられる。

速度制御で実験をした結果、小さな力で操作ができることがわかった。これにより、お年寄りなどの力が弱くなっている人が操作するのに大きな力を出す必要がなくなるので、負担が小さく考えられる。また、力の大きさの変化は、 $F_x$ 、 $F_z$  とともに小さいため代用特性である  $F_x/F_z$  の値の変動も小さい。

実験において歩行器は、スムーズに動作していたので  $F_x/F_z$  は代用特性として成立すると考えられ、この代用特性を用いての速度制御が確立できた。

#### 9.9 まとめ

今現在想定している幅広いターゲット層から、工学面、医学面、地域・環境面の視点から対象者の歩行能力レベルと歩行支援機の支援レベルの検討をする必要があることがわかった。

実験から、一定速で支援機が動くことにより、ハンドルを握り締めて大きな力を加え続ける必要がないために人が歩行するときのリズムの影響を受けにくく、ほぼ一定で力の大きさの変化が小さいことがわかった。これは、 $F_x/F_z$  の代用特性の値の変化が小さいため、速度変化が起こりにくいと考えられる。なんらかの原因で力が大きく変動したときは、速度が変化するために元の速度に戻そうとして人が力を加えたり減らしたりして調整をしていると考えられる。本研究の目的としても、歩行支援機が人間のすべてを捕うものではなく、人と機器とのコミュニケーションをしていくものである。

ハンドルに加えられる力の大きさは、センサとの接触の具合によって変化してしまう。実験においては、 $F_x$  方向に力を加えなくても、センサが力を検出してしまったり、斜め前方に力を加えたことで、 $F_x$  方向の力が検知されず制御ができなくなったりしたことがあったため、センサの取り付け方を変える必要がある。同時に、ハンドルがずれてしまったために、センサが力を検知できなくなったので、ハンドルの固定方法を考える必要がある。

$F_x/W$  の代用特性を用いたことで、 $F_x$ 、 $W$  とともに加えられる力のぶれが少なく、ばらつきが小さい。このことから、 $F_x$

／Fz 比による代用特性での制御で問題がないと考えられるが、さらにモニター試験等を繰り返し行う必要があると考えられる。

使用者が歩行支援機に上下、前後方向に加えた荷重を検知することで、前進に関してはどのように移動した以下を判断することができるになりスイッチレスになったが、バックやクラッチなどの操作の判断をすることで完全なスイッチレスをすることができるようになる。

## 9.9 A/D 変換プログラム

```

ERRORLEVEL      -207
LIST             P=18F8720,ST=OFF,R=DEC,N=125
INCLUDE          P18F8720.INC          __CONFIG __CONFIG1H, _OSCS_OFF_1H  &  _HS_OSC_1H
;_HSPLL_OSC_1H
__CONFIG __CONFIG2L, _BOR_OFF_2L & _PWRT_OFF_2L          & _BORV_42_2L

__CONFIG __CONFIG2H, _WDT_OFF_2H & _WDTPS_128_2H
__CONFIG __CONFIG3L, _MC_MODE_3L
__CONFIG __CONFIG3H, _CCP2MX_OFF_3H
__CONFIG __CONFIG4L, _STVR_ON_4L & _LVP_OFF_4L & _DEBUG_OFF_4L
__CONFIG __CONFIG5L, _CP0_OFF_5L & _CP1_OFF_5L & _CP2_OFF_5L          & _CP3_OFF_5L & _CP4_OFF_5L &
_CP5_OFF_5L & _CP6_OFF_5L  & _CP7_OFF_5L
__CONFIG __CONFIG5H, _CPB_OFF_5H          & _CPD_OFF_5H
__CONFIG __CONFIG6L, _WRT0_OFF_6L & _WRT1_OFF_6L          & _WRT2_OFF_6L & _WRT3_OFF_6L & _WRT4_OFF_6L
& _WRT5_OFF_6L & _WRT6_OFF_6L & _WRT7_OFF_6L
__CONFIG __CONFIG6H, _WRTC_OFF_6H & _WRTB_OFF_6H          & _WRD_OFF_6H

__CONFIG __CONFIG7L, _EBTR0_OFF_7L & _EBTR1_OFF_7L & _EBTR2_OFF_7L & _EBTR3_OFF_7L & _EBTR4_OFF_7L &
_EBTR5_OFF_7L & _EBTR6_OFF_7L & _EBTR7_OFF_7L
__CONFIG __CONFIG7H, _EBTRB_OFF_7H
;__CONFIG __CONFIG1H, _OSCS_OFF_1H"          & _OSC_HS_1H          ;_OSC_HSPLL_1H

;__CONFIG __CONFIG2L, _BOR_OFF_2L" & _PWRT_OFF_2L          & _BORV_42_2L
;__CONFIG __CONFIG2H, _WDT_OFF_2H"& _WDTPS_128_2H
;__CONFIG __CONFIG3L, _MODE_MC_3L "
;__CONFIG __CONFIG3H, _CCP2MUX_OFF_3H"
;__CONFIG __CONFIG4L, _STVR_ON_4L" & _LVP_OFF_4L          & _DEBUG_ON_4L
;__CONFIG __CONFIG5L, _CP0_OFF_5L" & _CP1_OFF_5L          & _CP2_OFF_5L          & _CP3_OFF_5L
;__CONFIG __CONFIG5H, _CPB_OFF_5H" & _CPD_OFF_5H
;__CONFIG __CONFIG6L, _WRT0_OFF_6L"          & _WRT1_OFF_6L          & _WRT2_OFF_6L          & _WRT3_OFF_6L
;__CONFIG __CONFIG6H, _WRTC_OFF_6H"          & _WRTB_OFF_6H          & _WRD_OFF_6H
;__CONFIG __CONFIG7L, _EBTR0_OFF_7L"          & _EBTR1_OFF_7L          & _EBTR2_OFF_7L          & _EBTR3_OFF_7L
;__CONFIG __CONFIG7H, _EBTRB_OFF_7H"
VER0 EQU '0'
VER1 EQU '0'
VER2 EQU '0'
VER3 EQU '0'
VER4 EQU '0'
VER5 EQU '0'
VER6 EQU '0'
VER7 EQU '1'
__IDLOCS __IDLOC0, VER0          ;          ;
__IDLOCS __IDLOC1, VER1          ;          ;

```

```

__IDLOCS __IDLOC2, VER2          ;
__IDLOCS __IDLOC3, VER3          ;
__IDLOCS __IDLOC4, VER4          ;
__IDLOCS __IDLOC5, VER5          ;
__IDLOCS __IDLOC6, VER6          ;
__IDLOCS __IDLOC7, VER7          ;
A      EQU 0                      ; RAM location in AccessRAM(BSRregister is ignored)
B      EQU 1                      ; RAM bank is specified by BSR register
      EEPROM_SCL      EQU 3          ; RC3 OUTPUT
      EEPROM_SDA      EQU 4          ; RC4 OUTPUT / INPUT
;-----
      SYUT_GX          EQU 0
      SYUT_GY          EQU 1
      SYUT_FX          EQU 2
      SYUT_FY          EQU 3
      SYUT_FZ          EQU 4
      SYUT_LONG        EQU 5
;-----
      SW_0             EQU 4          ; RE4 INPUT
      SW_1             EQU 5          ; RE5 INPUT
      SW_2             EQU 6          ; RE6 INPUT
      SW_3             EQU 7          ; RE7 INPUT
;-----
      OUT_ADD_FX        EQU 0          ; RF0 OUTPUT          LED
      OUT_ADD_FY        EQU 1          ; RF1 OUTPUT          LED
      OUT_ADD_FZ        EQU 2          ; RF2 OUTPUT          LED
      OUT_LONG          EQU 3          ; RF3 OUTPUT          LED
      OUT_MOTR_LOT      EQU 4          ; RF4 OUTPUT          LED
      OUT_MOTL_LOT      EQU 5          ; RF5 OUTPUT          LED
      LED_6             EQU 6          ; RF6 OUTPUT          LED
      LED_7             EQU 7          ; RF7 OUTPUT          LED
;-----
      LONG             EQU 0          ;RG0 INPUT
      MOTORR_ROT        EQU 1          ;RG1 INPUT
      MOTORL_ROT        EQU 2          ;RG2 INPUT
;-----
      GX               EQU 0          ;RH0 INPUT
      GY               EQU 1          ;RH1 INPUT
      FXR              EQU 2          ;RH2 INPUT
      FYR              EQU 3          ;RH3 INPUT
      FZR              EQU 4          ;RH4 INPUT
      FXL              EQU 5          ;RH5 INPUT
      FYL              EQU 6          ;RH6 INPUT
      FZL              EQU 7          ;RH7 INPUT
;-----
      MOTOR

      STACK_W          EQU 0x000      STACK_S
      EQU 0x002          STACK_B      EQU 0x004      STACK_F
                        EQU 0x006      STACK_PORTD    EQU 0x008
                        STACK_TRISD      EQU 0x00A      SUBWREG

```

```

EQU 0x00C          PCLATL          EQU 0x00E
TEMP_MAIN_A          EQU 0x010          TEMP_MAIN_B
EQU 0x012          ;
TEMP_MAIN_C          EQU 0x014          ;
TEMP_MAIN_D          EQU 0x016          ;
TEMP_SUB_A          EQU 0x018          ;
TEMP_SUB_B          EQU 0x01A          ;
TEMP_SUB_C          EQU 0x01C          ;
TEMP_SUB_D          EQU 0x01E          TIMER1_SUB
EQU 0x020          ;
TIMER1_HI          EQU 0x022          ;
TIMER1_LO          EQU 0x024          ;
PORTJ_TEMP          EQU 0x026          ;
KEY_NUMBER          EQU 0x028          ;
WAIT_00          EQU 0x02A          ;
WAIT_01          EQU 0x02C          ;
WAIT_02          EQU 0x02E          BCD_UP          EQU 0x030
;
BCD_HI          EQU 0x032          ;
BCD_LO          EQU 0x034          ;
BCDBIN_TEMP00          EQU 0x036          ;
BCDBIN_TEMP01          EQU 0x038          ;
BINARY_UP          EQU 0x03A          ;
BINARY_HI          EQU 0x03C          ;
BINARY_LO          EQU 0x03E          ;
TIMER_FLAG          EQU 0x040          ;
_LED_DECREMENT          EQU 0
SCAN_GX_RH0          EQU 0x042          ;
SCAN_GY_RH1          EQU 0x044          ;
SCAN_FXR_RH2          EQU 0x046          ;
SCAN_FYR_RH3          EQU 0x048          ;
SCAN_FZR_RH4          EQU 0x04A          ;
SCAN_FXL_RH5          EQU 0x04C          ;
SCAN_FYL_RH6          EQU 0x04E          ;
SCAN_FZL_RH7          EQU 0x050          ;
SCAN_LONG_RG0          EQU 0x052          ;
SCAN_MOTR_LOT_RG1          EQU 0x054          ;
SCAN_MOTL_LOT_RG2          EQU 0x056          ;
FX_ADD          EQU 0x058          ;
FY_ADD          EQU 0x05A          ;
FZ_ADD          EQU 0x05C          ;

          ORG 0x0000

NOP
INITIALIZE          ORG 0x0008

          HIGH_PRIORITY_INT
          GOTO

HIGH_PRIORITY_ISR          ORG
0x0018          LOW_PRIORITY_INT
          GOTO LOW_PRIORITY_ISR

```

```

HIGH_PRIORITY_ISR
    MOVWF STACK_W,A
    MOVFF STATUS,STACK_S
BSR,STACK_B
    MOVFF TRISD,STACK_TRISD
PIR1,TMR1IF,A
    BCF PIR1,TMR1IF,A
    CALL ISR_TMR1
TMR1IF_END
    MOVFF STACK_TRISD,TRISD
STACK_PORTD,PORTD
    MOVF STACK_W,W,A
    MOVFF STACK_S,STATUS
    MOVFF LOW_PRIORITY_ISR
    NOP
RETFIE
;
    RETURN
ISR_TMR1
    MOVLW B'11110000'
    MOVWF TMR1H,A
B'00010000'
    XORWF PORTG,F,A
TIMER_FLAG_LED_DECREMENT
    MOVLW 8
    XORWF TIMER1_SUB,W
STATUS,Z
    GOTO ISR_TMR1_INCREMENT_END
TIMER1_SUB
    MOVLW 1
    ADDWF TIMER1_LO,F
STATUS,C,A
TIMER1_HI,F
    INITIALIZE
        MOVLW B'00000110'
        MOVLW B'00000000'
        MOVWF INTCON2,A
        MOVLW B'11111111'
        MOVWF PORTA,A
        MOVLW B'11111111'
        MOVWF PORTB,A
    MOVLW B'00000000'
        MOVLW B'00000000'
        MOVWF PORTC,A
    MOVLW B'00010000'
        MOVWF PORTD,A
    MOVLW B'00000000'
        MOVWF PORTE,A
        MOVLW B'11111111'

```

```

MOVWF PORTG,A
MOVLW B'11111111'
MOVWF PORTH,A
MOVLW B'00000000'
MOVWF PORTJ,A
MOVLW B'11111111'
MOVWF TRISA,A
MOVLW B'11111111'
MOVWF TRISB,A
MOVLW B'11111101'
MOVWF TRISC,A
MOVLW B'00000000'
MOVWF TRISD,A
MOVLW B'11111100'
MOVWF TRISE,A
MOVLW B'00000000'
MOVWF TRISF,A
MOVLW B'11101111'
MOVWF TRISG,A
MOVLW B'11111111'
MOVWF TRISH,A
MOVLW B'00001000'
MOVWF TRISJ,A
LFSR FSR0,0x000

INIT_CLEAR_ALL_BANK CLR POSTINC0
MOVLW 0x06
MOVWF FSR0H
CPFSEQ FSR0H,A
GOTO INIT_CLEAR_ALL_BANK MOVWF B'11000000'

MOVWF TMR1H,A
CLRF TMR1L,A
MOVLW B'00001111'
MOVWF T1CON,A
MOVLW 21
MOVWF SPBRG1,A
MOVLW B'00100100'
MOVWF TXSTA1,A
MOVLW B'10010000'
MOVWF RCSTA1,A
MOVLW 21
MOVWF SPBRG2,A
MOVLW B'00100100' MOVWF TXSTA2,A

MOVLW B'10010000'
MOVWF RCSTA2,A
CALL WAIT_20MS
CALL WAIT_5MS
CALL WAIT_100US
CALL WAIT_100US

```

```

                                NOP
                                MOVLW B'00000000'
                                MOVWF PIR1,A
MOVWLW B'00000001'
                                MOVWF PIE1,A
                                MOVLW B'11000000'
MOVWF INTCON,A
MAIN
                                NOP
                                CALL AD_CONVERT
                                RLCF          SCAN_GX_RH0,F
                                BCF          SCAN_GX_RH0,0
                                BTFSC       PORTH,0,A
                                BSF         SCAN_GX_RH0,0
                                MOVLW B'10000000'
                                XORWF      SCAN_GX_RH0,W
                                BTFSS      STATUS,Z,A
                                GOTO       KEY_GX_RH0_END
                                MOVLW 0
                                MOVWF     KEY_NUMBER,A
                                                RLCF          SCAN_GY_RH1,F
                                                BCF          SCAN_GY_RH1,0
                                BTFSC     PORTH,1,A
                                BSF       SCAN_GY_RH1,0
                                MOVLW B'10000000'
                                                XORWF
SCAN_GY_RH1,W
                                BTFSS     STATUS,Z,A
                                GOTO      KEY_GY_RH1_END
                                MOVLW 1
MOVWF KEY_NUMBER,A
                                                RLCF          SCAN_FXR_RH2,F
                                BCF       SCAN_FXR_RH2,0
                                BTFSC     PORTH,2,A
                                BSF       SCAN_FXR_RH2,0
                                                MOVLW B'10000000'
                                                XORWF      SCAN_FXR_RH2,W
                                BTFSS     STATUS,Z,A
                                GOTO      KEY_FXR_RH2_END
                                                MOVLW 2
                                MOVWF     KEY_NUMBER,A
KEY_FXR_RH2_END
                                RLCF      SCAN_FYR_RH3,F
                                                BCF          PORTH,3,A
                                SCAN_FYR_RH3,0
                                                BTFSC       SCAN_FYR_RH3,0
                                MOVLW B'10000000'
                                                XORWF
SCAN_FYR_RH3,W
                                                BTFSS      STATUS,Z,A
                                                GOTO      KEY_FYR_RH3_END
                                MOVLW 3
MOVWF KEY_NUMBER,A
                                                RLCF          SCAN_FZR_RH4,F
                                BCF       SCAN_FZR_RH4,0

```

```

BTFSC          PORTH,4,A                      BSF
SCAN_FZR_RH4,0                      MOVLW  B'10000000'
                                XORWF  SCAN_FZR_RH4,W
BTFSS          STATUS,Z,A                      GOTO
KEY_FZR_RH4_END                      MOVLW  4
                                MOVWF  KEY_NUMBER,A
KEY_FZR_RH4_END
RLCF          SCAN_FXL_RH5,F                      BCF
SCAN_FXL_RH5,0                      BTFSC          PORTH,5,A
                                BSF          SCAN_FXL_RH5,0
                                MOVLW  B'10000000'
                                XORWF
SCAN_FXL_RH5,W                      BTFSS          STATUS,Z,A
                                GOTO          KEY_FXL_RH5_END
                                MOVLW  5
MOVWF  KEY_NUMBER,A                      KEY_FXL_RH5_END
                                RLCF          SCAN_FYL_RH6,F
                                BCF          SCAN_FYL_RH6,0
BTFSC          PORTH,6,A                      BSF
SCAN_FYL_RH6,0                      MOVLW  B'10000000'
                                XORWF  SCAN_FYL_RH6,W
BTFSS          STATUS,Z,A                      GOTO
KEY_FYL_RH6_END                      MOVLW  6
                                MOVWF  KEY_NUMBER,A
                                RLCF          KEY_FYL_RH6_END
SCAN_FZL_RH7,F                      BCF          SCAN_FZL_RH7,0
                                BTFSC          PORTH,7,A
                                BSF          SCAN_FZL_RH7,0
                                MOVLW  B'10000000'
                                XORWF  SCAN_FZL_RH7,W
                                BTFSS          STATUS,Z,A
                                GOTO          KEY_FZL_RH7_END
                                MOVLW  7
                                MOVWF  KEY_NUMBER,A
                                RLCF          KEY_FZL_RH7_END
                                SCAN_LONG_RG0,F                      BCF
SCAN_LONG_RG0,0                      BTFSC          PORTG,0,A
                                BSF          SCAN_LONG_RG0,0
                                MOVLW  B'10000000'
                                XORWF
SCAN_LONG_RG0,W                      BTFSS          STATUS,Z,A
                                GOTO          KEY_LONG_RG0_END
                                MOVLW  8
MOVWF  KEY_NUMBER,A
KEY_LONG_RG0_END
RLCF          SCAN_MOTR_LOT_RG1,F                      BCF
SCAN_MOTR_LOT_RG1,0                      BTFSC          PORTG,1,A
                                BSF          SCAN_MOTR_LOT_RG1,0
                                MOVLW  B'10000000'
                                XORWF
SCAN_MOTR_LOT_RG1,W                      BTFSS          STATUS,Z,A
                                GOTO          KEY_MOTR_LOT_RG1_END
                                MOVLW  9
                                MOVWF  KEY_NUMBER,A
                                RLCF          SCAN_MOTL_LOT_RG2,F
BTFSC          SCAN_MOTL_LOT_RG2,0                      BTFSC

```



```

PORTG2,A                                     BSF
SCAN_MOTL_LOT_RG2,0                         MOVLW  B'10000000'
                                           XORWF  SCAN_MOTL_LOT_RG2,W
BTFSS   STATUS,Z,A                           GOTO
KEY_MOTL_LOT_RG2_END                         MOVLW  10
                                           MOVWF  KEY_NUMBER,A
                                           CLRF   TIMER1_HI,A
CLRF    TIMER1_LO,A
KEY_MOTL_LOT_RG2_END
                                           BTFSS  PIR1,RC1IF,A
MOVF    RCSTA1,W,A                           ANDLW
B'00000110'                                  XORLW  B'00000000'

ADD
                                           MOVF   SCAN_FXR_RH2,0
                                           ADDWF  SCAN_FXL_RH5,0
                                           MOVWF  FX_ADD,0

                                           MOVF   SCAN_FYR_RH3,0
                                           ADDWF  SCAN_FYL_RH6,0
                                           MOVWF  FY_ADD,0

                                           MOVF   SCAN_FZR_RH4,0
                                           ADDWF  SCAN_FZL_RH7,0
                                           MOVWF  FZ_ADD,0

OUT
                                           MOVF   FX_ADD,0
                                           MOVWF  PORTF,0
                                           CLRF  PORTF
                                           MOVF  FY_ADD,0
                                           MOVWF PORTF,1
                                           CLRF  PORTF
                                           MOVF  FZ_ADD,0
                                           MOVWF PORTF,2
                                           CLRF  PORTF
                                           MOVF  SCAN_LONG_RG0,0
                                           MOVWF PORTF,3
                                           CLRF  PORTF
                                           MOVF  SCAN_MOTR_LOT_RG1,0
                                           MOVWF PORTF,4
                                           CLRF  PORTF
                                           MOVF  SCAN_MOTL_LOT_RG2,0
                                           MOVWF PORTF,5
                                           CLRF  PORTF

                                           CLRF  PORTD
                                           MOVF  SCAN_GX_RH0,0
                                           MOVWF PORTD,0
                                           CLRF  PORTD
                                           MOVF  SCAN_GY_RH1,0

```

```

MOVWF PORTD,1
CLRF PORTD
MOVF FX_ADD,0
MOVWF PORTD,2
CLRF PORTD
MOVF FY_ADD,0
MOVWF PORTD,3
CLRF PORTD
MOVF FZ_ADD,0
MOVF PORTD,4
CLRF PORTD
MOVF SCAN_LONG_RG0,0
MOVWF PORTD,5
CLRF PORTD

BTFSC STATUS,Z,A

COMMAND_END ;
GOTO MAIN
WAIT_50US MOVLW 62 CALL
WAIT_COMMON_US WAIT_100US MOVLW 125
CALL WAIT_COMMON_US
WAIT_COMMON_US
NOP
DECFSZ WAIT_00,F GOTO
WAIT_COMMON_US RETURN
WAIT_5MS
MOVLW 30
MOVWF WAIT_00 CLRF WAIT_01
GOTO WAIT_COMMON_MS
WAIT_10MS
MOVLW 60
MOVWF WAIT_00
CLRF WAIT_01
GOTO WAIT_COMMON_MS
WAIT_20MS
MOVLW 120
MOVWF WAIT_00
CLRF WAIT_01
GOTO WAIT_COMMON_MS
WAIT_COMMON_MS
NOP
DECFSZ WAIT_01,F GOTO
WAIT_COMMON_MS DECFSZ WAIT_00,F
GOTO WAIT_COMMON_MS
RETURN WAIT_250MS
MOVLW 6
MOVWF WAIT_00 CLRF WAIT_01
CLRF WAIT_02
GOTO WAIT_COMMON_SEC WAIT_500MS
MOVLW 12
MOVWF WAIT_00 CLRF WAIT_01
CLRF WAIT_02

```

```

                GOTO    WAIT_COMMON_SEC                WAIT_1S
                MOVLW   24
MOVWF    WAIT_00                                CLRF    WAIT_01

                CLRF    WAIT_02
GOTO    WAIT_COMMON_SEC                WAIT_COMMON_SEC
                NOP
                DECFSZ  WAIT_02,F
GOTO    WAIT_COMMON_SEC
                DECFSZ  WAIT_01,F
                GOTO    WAIT_COMMON_SEC
                DECFSZ  WAIT_00,F
GOTO    WAIT_COMMON_SEC                                RETURN

AD_CONVERT ;
                MOVLW   B'00001110'
                MOVWF   ADCON1,A
MOVWLW   B'00000111'
                MOVWF   ADCON2,A
MOVWLW   B'00000001'
                MOVWF   ADCON0,A
CALL     WAIT_50US                                BSF
ADCON0,GO,A
                BTFSC  ADCON0,GO,A
GOTO    $-2

BIN_2_BCD6 ;
BIN_2_BCD6_START
                MOVLW   1/24
MOVWF   BCDBIN_TEMP01,B                                CLRF    BCD_UP,B
                CLRF    BCD_HI,B
                CLRF    BCD_LO,B
BCF     STATUS,C,A                                BIN_2_BCD6_LOOP24
                RLCF    BINARY_LO,F,B
RLCF    BINARY_HI,F,B                                RLCF
BINARY_UP,F,B                                BCD_LO,F,B
                RLCF    BCD_HI,F,B
                RLCF    BCD_UP,F,B
                MOVLW   0x03                                ADDWF
BCD_LO,W,B                                BTFSC
BCDBIN_TEMP00,3,B
                MOVWF   BCD_LO,B                                MOVLW
0x30                                ADDWF   BCD_LO,W,B
                MOVWF   BCDBIN_TEMP00,B
                BTFSC  BCDBIN_TEMP00,7,B                                MOVWF   BCD_LO,B
                MOVLW   0x03
                ADDWF   BCD_HI,W,B
MOVWF   BCDBIN_TEMP00,B                                BTFSC
BCDBIN_TEMP00,3,B                                MOVWF   BCD_HI,B
                MOVLW   0x30
ADDWF   BCD_HI,W,B                                MOVWF

```

```

BCDBIN_TEMP00,B                                BTFSC                                BCDBIN_TEMP00,7,B
                                                ;
MOVLW 0x03                                     ADDWF
BCD_UP,W,B                                     MOVWF BCDBIN_TEMP00,B
                                                BTFSC                                BCDBIN_TEMP00,3,B
MOVWF BCD_UP,B                                 MOVLW 0x30
                                                ADDWF BCD_UP,W,B
MOVWF BCDBIN_TEMP00,B                         BTFSC
BCDBIN_TEMP00,7,B                             MOVWF BCD_UP,B
DECFSZ BCDBIN_TEMP01,F,B                     GOTO
BIN_2_BCD6_LOOP24
RLCF BINARY_LO,F,B
RLCF BINARY_HI,F,B                            RLCF
BINARY_UP,F,B
RLCF BCD_LO,F,B
RLCF BCD_HI,F,B
RLCF BCD_UP,F,B
RETURN
end

```

## 5.2 PWMプログラム

```

#include <18F8720.h>
#include <stdio.h>
#include <math.h>

#fuses BORV45,NOBROWNOUT,PUT
#fuses CCP2C1
#fuses NODEBUG,NOLVP,NOSTVREN
#fuses NOPROTECT,NOCPD,NOCPB
#fuses NOWRT,NOWRTD,NOWRTB,NOWRTC,NOEBTR,NOEBTRB

#use delay(CLOCK=4000000)

#use fast_io(b)
#use standard_io(f)
#use standard_io(e)

void main()
{
    int i;
    int fxr;
    int fyr;
    int fzr;
    int fxl;
    int fyl;
    int fzl;
    int fx;
    int fy;

```

```

int fz;
int a,c;
int avefx;
int avefz;

for(i=1;;i++)
{
    a=0;
    c=0;

    fxr = pin_f0;
    fyr = pin_f1;
    fzr = pin_f2;
    fxl = pin_f3;
    fyl = pin_f4;
    fzl = pin_f5;

    fx = fxr + fxl;
    fy = fyr + fyl;
    fz = fzr + fzl;

    avefx = fx+a/i;
    avefz = fz+c/i;

    a=avefx;
    c=avefz;

    if(c <= 3 && 2<c)
    {
        if(a > 3)
        {

            long duty;

            output_low(pin_e0);
            output_high(pin_e1);

            setup_CCPI(CCP_PWM);
            setup_timer_2(T2_DIV_BY_1,0xFF,1);
            setup_CCP2(CCP_PWM);

            for(;;)
            {

                set_pwm1_duty(duty++);
                set_pwm2_duty(duty++);

                delay_ms(100);

                if(set_pwm1_duty(100))
                {
                    break;

```

```

        }

        else
        {
        }
    }
}

else
{
    delay_ms(100);
}
}

if(c > 3)
{
    if(a > 3)
    {
        long duty;

        setup_ccp1(CCP_PWM);
        setup_timer_2(T2_DIV_BY_1,0xFF,1);

        setup_ccp2(CCP_PWM);

        duty=0;

        for(;;)
        {
            set_pwm1_duty(duty++);
            delay_ms(100);

            set_pwm2_duty(duty++);

            if(set_pwm1_duty(100))
            {
                break;
            }

            else
            {
            }
        }
    }
}

else
{
    long duty;

    setup_ccp1(CCP_PWM);
    setup_ccp2(CCP_PWM);
    setup_timer_2(T2_DIV_BY_1,0xFF,1);
}
}

```

```

duty=0;

for(;;)
{
    set_pwm1_duty(duty++);
    delay_ms(100);
    set_pwm2_duty(duty++);

    if(set_pwm1_duty(100))
    {
        break;
    }

    else
    {
    }
}
}

```

## 10. 移動動作と制御系システムの検証

### 10.1 移動動作解析と測定

歩行支援機の制御を行うにあたり、歩行支援機と非支援者の速度は非常に重要なポイントである。既存の歩行機と試作機を用いて人と歩行機の速度関係を調べることにする

#### (1) 実験装置

実験装置の構成を Fig.4.1 に示す。被験者の左右側方 5m の地点に 2 台の CCD カメラを被験者に対し 60 度になるように設置し、各関節の角度・距離などの座標データを PC で記録する。(Detect 社 3 次元画像解析装置 Dipp Motion XD)

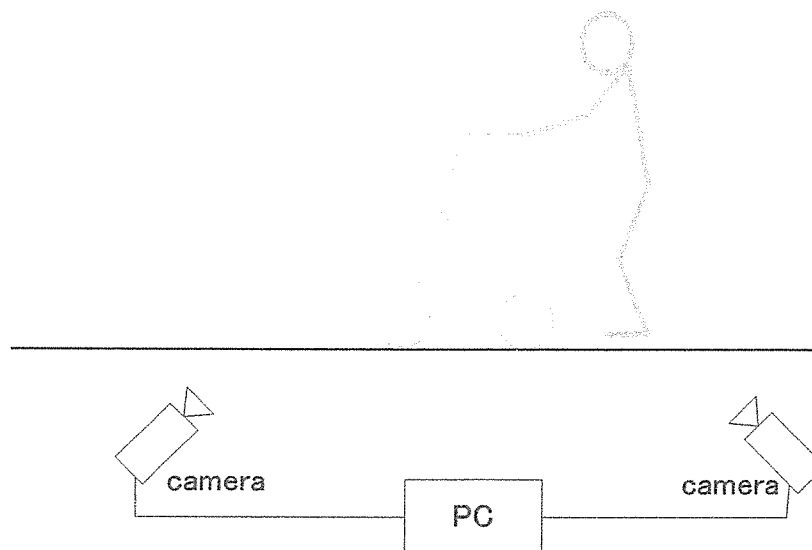


Fig.10.1 Configuration of Operating System

#### (2) 使用する歩行器の選定

計測に用いた歩行器はコスモ株式会社の販売するコスモネティーを選定した。これは、ホイールベース 500mm, 後ト

レッド幅 600[mm], 前トレッド幅 500[mm], 車輪直径 200[mm]としたコンセプトデザイン<sup>(2)</sup>に近かったものである. 各部の寸法を Fig.4.2 に示す. スチール製で重量は 8.8[kg]である.

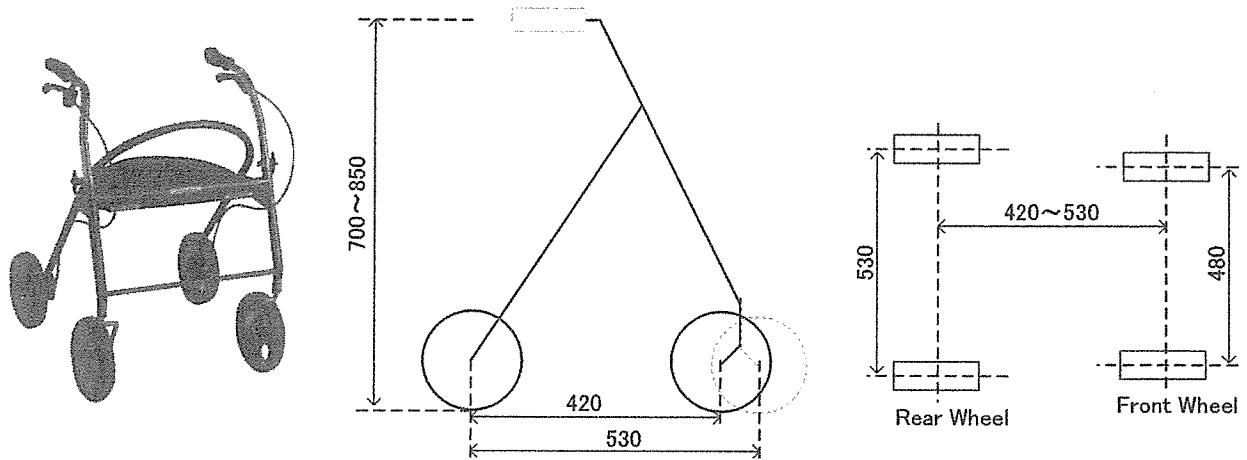


Fig. 10.2 Walker

(3) 実験方法

Fig.10.3 に示すように, 被験者の①左肩峰点②左肘関節③左手首④左大転子点⑤左膝関節⑥左踝(踵)⑦つま先と, 支援機の前輪, 後輪, ハンドルの3箇所を計測した. 速度を調べるために歩行機のハンドル部と④左大転子点をトレースすることにより速度を求めることとした.

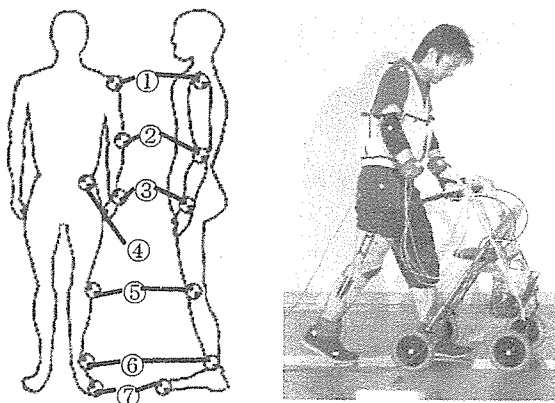


Fig. 10.3 The Marker Position of 3-dimensional Measurement & Senior Citizen Experience Kit

Tab.10.1 Mean velocity

Person A of Mean velocity[m/s]		
subject	person	supporting machine
1	0.40	0.40
2	0.44	0.39
3	0.41	0.41

Person B of Mean velocity[m/s]		
subject	person	supporting machine
1	0.53	0.56
2	0.55	0.55
3	0.62	0.72

Person C of Mean velocity[m/s]		
subject	person	supporting machine
1	0.36	0.30
2	0.33	0.34
3	0.47	0.41



#### (4) 考察

実験結果より、歩行機と使用者の速度は周期的に変化しているが、双方の速度が変化の周期が同期する事はない。また、人の速度変化の方が変化の波が大きく、歩行機は変化がすくない。

Tab10.2 は標準偏差を算出した結果である。これからも分かるように歩行機の方が、ばらつきが小さい事が示されている。また、歩行機の速度は一定の速度に達すると変化の幅は小さくなり、安定する傾向が見られる。これは、歩行機に頼って移動するときに、歩行機の速度が急激に変化していると慣性力が大きくなるため、制御しにくくなるのを未然に防いでいるためと思われる。

このことから、個人差はあるもののある一定の速度にしようとしていると考えられる。これらにより、安定走行時には一定の速度で制御を行う制御が有効ではないかと考えられる。

Tab.10.2 Standard deviation of velocity

Person A		Person B	
person	supporting machine	person	supporting machine
0.23	0.15	0.32	0.32
0.24	0.19	0.21	0.21
0.23	0.20	0.26	0.25

Person C	
person	supporting machine
0.24	0.13
0.25	0.17
0.26	0.16

## 10. 2. ハンドル部の荷重センサーシステム

### (1) 電動モータによる動力アシストの制御パラメータ

制御パラメータの代表的なシステムとして、

- ①歩行支援機の前・後方への動力支援・負荷制御の場合は前進力  $F_x$  のみでよい。しかし、速度制御の場合は適合する代用特性をハンドル部からの情報のセンスにより得る必要がある。進行方向の力  $F_x$  と鉛直下向きにかかる力  $F_z$  より求める力  $F$  のベクトル方向を制御パラメータとして用いることを検討している。実験結果よりこの荷重マトリックス ( $F_x$ ,  $F_z$ ) と角度  $\theta (= \tan(F_z/F_x))$  を歩行支援機の色度制御の代用特性とする。
- ②歩行支援機の旋回に関するセンスは、 $F_x$  と  $F_y$  の合力のベクトル方向やハンドルに加わるトルクをセンスする方法がある。

### (2) ハンドル荷重検知装置の仕様

ハンドル部に加わる荷重の検出は、ハンドル部と本体フレームの間に3次元の前後、上下、左右方向の荷重のセンシングにより行う。その際の必要要件を列記する。

- ①ハンドルは、ユニバーサルデザインの導入により、肘掛け式と握りハンドル式の両方に適合する必要がある。ハンドルの使用方法により、被歩行支援者によるハンドルへの荷重のパターンは変化する。どのような方式でもハンドル全体での3方向の荷重を同じように測定できること。
- ②特に、前後に荷重位置が変わるため、左右方向軸廻りのモーメントに対して、センサー兼ハンドル支持部は、大きなねじり回転剛性を持つこと。
- ③上下荷重は、前後・左右方向よりも4~10倍大きいことから、その大きさに適合する感度と精度を持つこと。
- ④センサーの数は極力少ない方式がよい。
- ⑤センサーの配置は、同時にハンドルと本体フレーム間の支持点でもあり、信頼性も必要
- ⑥ハンドルに加わる被支援者のハンドル荷重は、左右別々に測定する必要はない。まとめて、3方向のハンドルにかかる全荷重を測定できればよい。
- ⑦測定信号は、測定データ処理のCPUチップでデジタル化され、メインCPUに送られること。
- ⑧ハンドル支持部兼センサー部は、原則全天候型



## 10.3 歩行支援機のアシスト方法について

### 10.3.1 統合モード

また、歩行支援機を制御するにあたり次のモードに分け、制御を行なう。場合によっては、制御系のカスタマイズのために下記の各モード毎にモジュールとしての最適化が図られ、これらの統合モードとして主コンピュータ制御部の調整が行われるものとする。

- (1) 前（後）進モード（含、極低速）
- (2) 登降坂モード
- (3) 段差乗越しモード（含、凸凹路）
- (4) 旋回モード
- (5) 自転モード
- (6) 停止モード（含、非常停止）

の6つのモードに分けて歩行支援機の制御を行なう。

前（後）進モードは、平坦路をハンドル部のセンサー信号により前（後）進方向に直進するように動力支援を行う。登降坂モードは、歩行支援機自身の運動性能に対する重力の加速度をキャンセルするような制御を行い、それ以外は原則平坦路の前（後）進モードに準じる。段差乗越しモードは、その段差等の通過時の衝撃を緩和するような通過モードであり、前輪通過後に被支援者の足の送りが段差等のバリアーに到達する際の「つまづき」への警告モードも含める。旋回モードは、機械的な操舵装置も用いるが、原則として、自転モードと前（後）進モードを組合わせたものである。自転モードは、左右の駆動車輪を逆転することで、自転するものである。狭い通路での方向転換やエレベータ内での自転を想定している。停止モードは、被支援者が移動を中止する際のモードであるが、危険防止のための非常停止を含む。

### 10.3.2 前進モードの動力支援方式

#### (1) 速度制御と負荷制御

機器に加わる力は歩く周期で増減しているため、常に一定の力が加わっているわけではない。図 10.7 に示すように、歩行の左右交互移動による1サイクル中での歩行支援機に加わる前進押力  $F_x$  と上下支持力  $F_z$  は、常に変化している。足の運びによってその押し力や支持力が変化することは報告している<sup>(3)</sup>。

従来は、 $F_x$  の値により電動モータの補助駆動力を決めていたが、この方式では負荷制御のため物理的に適正値を見いだそうとする方式であり、人間の感性とは一致しない。人間が（歩行）移動する際は、ビジュアルな情報を頼りに移動している場合が多くその寄与も大きい。従って、目の前の移動目標にどのくらい早く近づくかが一義的であり、移動のためにどのくらい負荷すなわち力を要するのかが予測できない。常に押してみても抵抗力が大きいとそれに対抗しようとする頭脳の判断によって間接的に移動を自分の意志に適合させようとする。すなわち、歩行移動において、速度制御が人間工学的にも直接的で分かりやすく、優れている。

ここで、被支援者がどのような速度で歩きたいのかを操作スイッチレスでどのように判断するかが大きな課題となる。歩行支援を受ける人は、余裕がない状態で歩行動作に集中しながら歩くことになるので、その際歩くことを中断して操作スイッチなり、レバーを動かすことは大きなリスクを伴う。

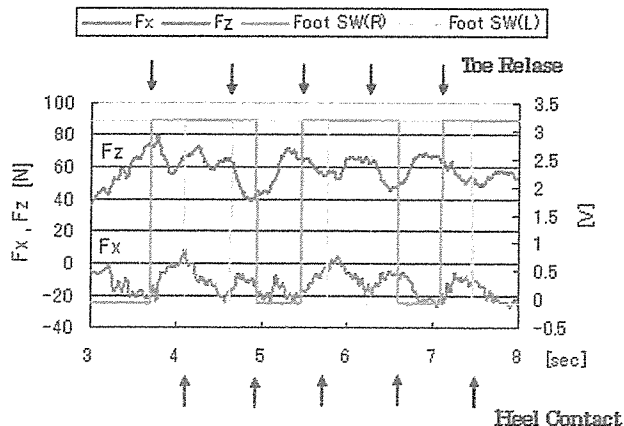


Fig. 10.7 Load on the handle at walking pattern

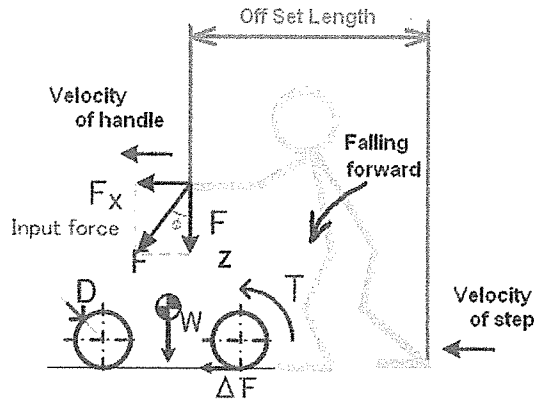


Fig. 10.8 Off set value between handle and step

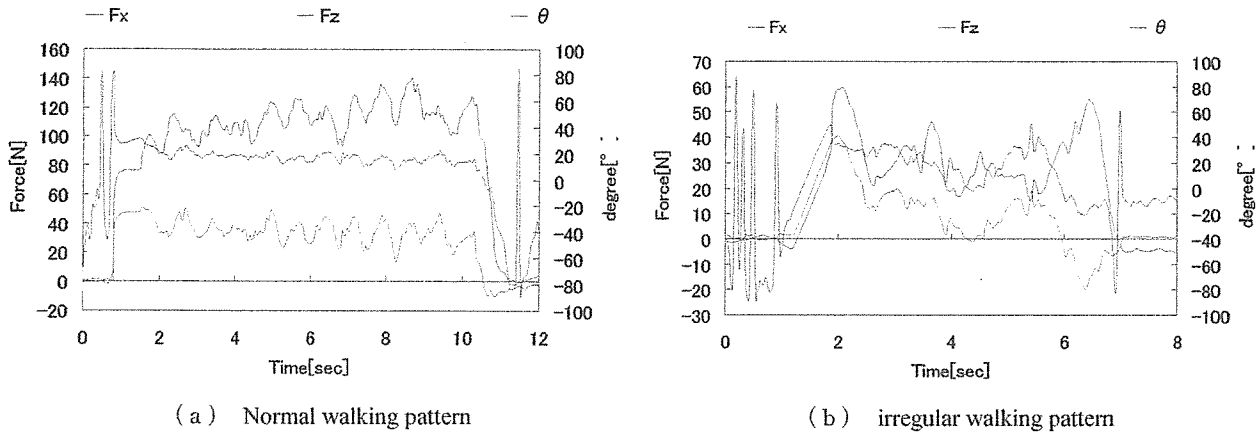


Fig. 10.9 Parameters of the handle load sensors

(2) 速度制御の代用特性について

我々の提案では、この被支援者の希望する歩行速度を設定する代用特性として、 $F_x$  と  $F_z$  の比率（例えば、 $F_x/F_z$  比、 $\tan \phi$  など）で設定することとする。前進押し力  $F_x$  の大きさのみで歩行支援機の駆動力または歩行速度を決めると、比較的歩行能力が高い人が軽いタッチですばやい歩行を望むと歩行支援機はゆっくりした動きとなり鈍重に感じる。また、逆に  $F_x$  の大きな押し力が加わった場合を考える。図 10.8 に示すように、比較的脚力が弱い人が、斜め前方方向に自分の体重を大きくかけ、何とか前進する時は、上下に体重をかけたハンドル支持点は足が動く速度よりも速く前方に移動し身体の前方への傾斜が更に増加し、いわゆる前のめり状態となり、不安定な姿勢となる。 $F_x$  の大きさにより設定された速度についてゆけるだけの脚力を持っている人であれば問題ないが、脚力により  $F_x$  の値に対する速度設定を適正に変化させる必要がある。その脚力のパラメータとして上下方向の体重支持力  $F_z$  を採用し、脚力を考慮した押し力： $F_x/F_z$  比を代用特性とする。図 10.9 にハンドル部に加わる前（後）方押し力  $F_x$  と上下方向体重支持力  $F_z$  および  $\theta = \tan^{-1}(F_x/F_z)$  の測定結果を示します。通常の歩行パターン(a)では、右左の歩行サイクルにより  $F_x$  や  $F_z$  は繰返しの変動がみられますが、 $\theta$  はほぼ一定の安定した値を示しています。速度制御の代用特性としては良好な特性といえます。(b)パターンでは、歩行時の歩行支援機への力のかけ方が大幅にちがう歩き方であればその変化に従って  $\theta$  の値が変動し、上下支持力が小さく歩行支援機に頼らない状態では前方に押し力が大きければ、目標速度の増加のため  $\theta$  値が増大するが、上下支持力が大きく歩行支援機に体重を預けている時は前方押し力が大きくとも速度設定は一定のままとなっており、安全な速度制御がなされているのが分かる。

10.3.3 上り坂と下り坂での歩行支援

前述の負荷制御方式では、勾配による重力の分力の負荷がハンドルに加えた被支援者の信号に加えて考える必要がある。上り坂では、平坦路より意識的に前方への押し力を加える必要がある。また、下り坂は特に歩行支援機にとって苦手な状況である。下りの重力分力により支援機が前方に加速され、図 10.7 のようなハンドルと足とのオフセット量が大きくなり、前のめりの転倒の危険が生じる。常に、後方への押し力により、制動力となる逆トルクを配慮する必要がある。

勾配による重力の分力をその路面状況によりハンドル操作で調整する動作を被支援者に求めることは難しい。速度制御