

Stanowisko
ZB2

Model zwijarki bębnowej dwusilnikowej

Dokumentacja techniczno-ruchowa

KSIS-PP
Poznań 2009

I. OPIS OGÓLNY. KONFIGURACJA I UŻYTKOWANIE

Podstawowe elementy składowe stanowiska ZB2:

- 1) fizyczny model zwijarki bębnowej z interfejsem analogowo-cyfrowym przystosowanym do współpracy z kartą I/O obsługiwaną przez komputer PC (wejścia i wyjścia analogowe $\pm 10V$, wejścia i wyjścia cyfrowe),
- 2) karta I/O (wejścia-wyjścia): PCI-DAS 1602/12,
- 3) komputer PC ze środowiskiem graficznym VisSim i nakładką RealTimePRO.

Zasada działania stanowiska.

Stanowisko ZB2 stanowi fizyczny model mechanizmu zwijarki, którego podstawowym zadaniem jest wybieranie i stabilizacja pozycji tzw. zwisu liny obciążonej wiszącą masą (widok ogólny modelu zwijarki ilustruje Rys. 1). Możliwa jest także realizacja bardziej złożonych zadań sterowania polegających na odtwarzaniu zadanej trajektorii położenia masy. Problem wybierania nadmiaru zwisającej taśmy pojawia się np. w przemyśle gumowym podczas procesu produkcyjnego opon samochodowych, a ściślej w procesie naklejania zbrojenia na taśmę przewijanej gumy z następującym po nim etapem zwijania prefabrykatu na bęben odbiorczy. Zadanie wybierania zwisu z określoną jakością dynamiczną i statyczną może okazać się problemem nietrywialnym w dwóch zasadniczych powodów:

- zmiennego promienia walca aktualnie zwiniętej taśmy na bębnie odbiorczym oraz
- zmiennej i często nieprzewidywalnej zmiany prędkości podawania taśmy przed wybraniem.

Wspomniane powody czynią rozważany proces **niestacjonarnym systemem dynamicznym** (promień walca jest parametrem obiektu) poddanym oddziaływaniu **zmiennego zakłócenia**.

Na stanowisku ZB2 wybieranie - a ogólniej zmiana długości zwisu - jest realizowana za pomocą bębna napędzanego silnikiem DC z przekładnią redukcyjną. Silnik S1 pracuje w obwodzie podrzędnej regulacji prędkości z regulatorem PID i sprzężeniem zwrotnym zrealizowanym w oparciu o sygnał z czujnika obrotowo-impulsowego (wykorzystano sterownik i wzmacniacz mocy firmy *miControl/Dunkermotoren BGE 3515*). Sygnałem wejściowym do obwodu podrzędnego jest wartość zadana prędkości silnika (sygnał 'uS1' z TABELI 4) - zatem napęd bębna jest sterowany prędkościowo (por. Rys 3.). Bieżąca pozycja zwisu (pozycja masy obciążającej) jest mierzona za pomocą potencjometrycznego czujnika linkowego umieszczonego na górnej podstawie zwijarki z końcem linki przymocowanej do masy obciążającej (bezwzględny pomiar pozycji).

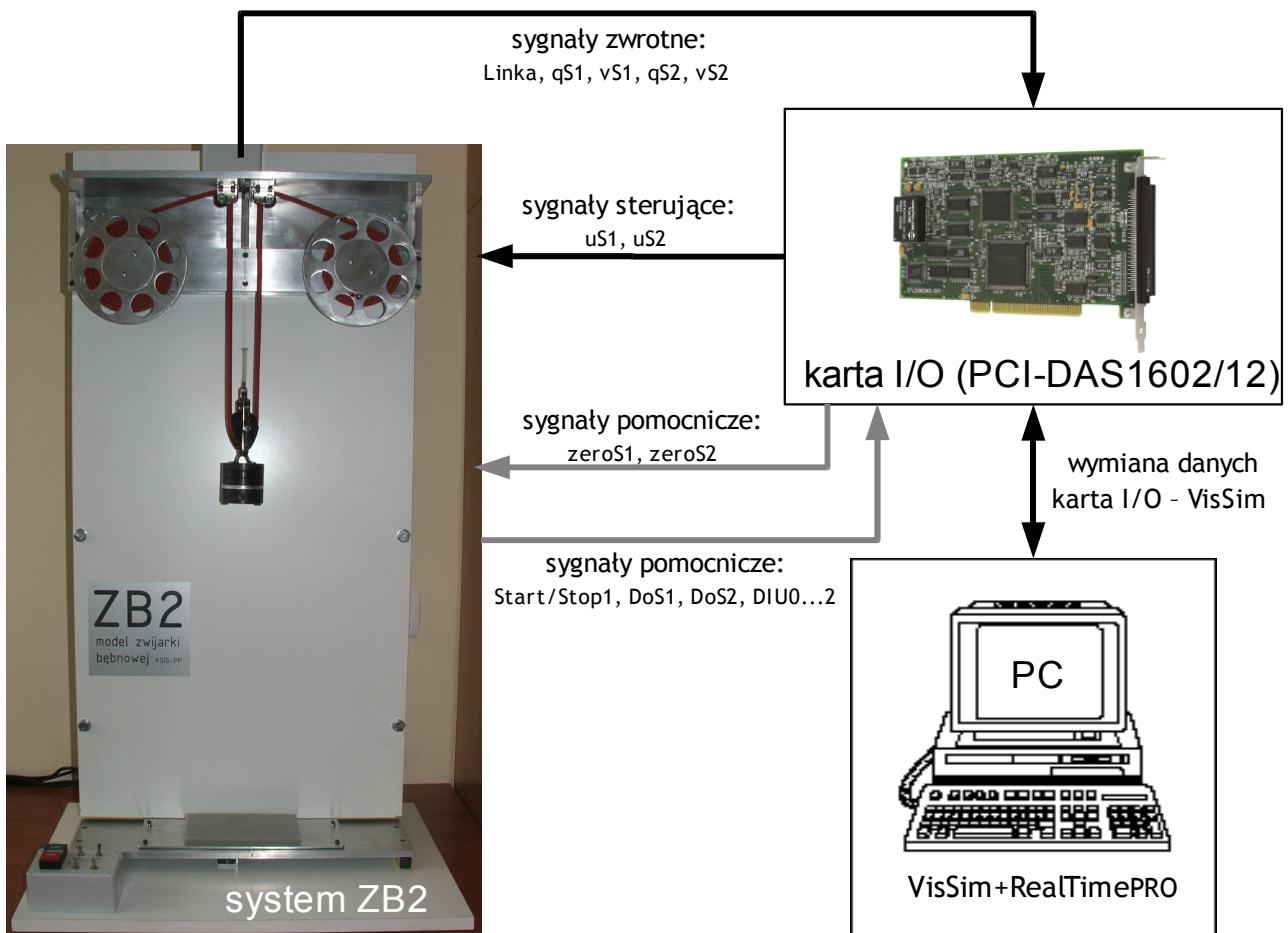
Sygnal pozycji zwisu (oznaczony jako 'Linka' w TABELI 4) jest podstawowym sygnałem wykorzystywanym w pętli sprzężenia zwrotnego obwodu regulacji nadrzędnej realizowanej w środowisku VisSim na komputerze PC. Projektowanie zewnętrznej pętli regulacyjnej stanowi zasadniczy cel pracy na stanowisku ZB2. Połączenie sygnałów wejściowych i wyjściowych zwijarki ZB2 ze środowiskiem VisSim zapewnia karta I/O zainstalowana w komputerze PC i współpracująca ze środowiskiem VisSim+RealTimePRO w czasie rzeczywistym.

Drugi silnik (S2) napędzający bęben dodatkowy stanowi programowalne źródło zakłócenia pozycji zwisu. Ruch bębna dodatkowego emuluje efekt podawania taśmy/liny ze zmienną prędkością, która w praktyce wynika z aktualnego stanu procesu produkcyjnego poprzedzającego wybieranie zwisu. Sterowanie silnikiem zakłócającym odbywa się analogicznie jak w przypadku silnika bębna wybierającego - ma charakter prędkościowy. Programową zmianę zakłócenia realizuje się poprzez generowanie (również w środowisku VisSim) i podawanie w torze otwartym wybranego profilu zmian prędkości na wejście sygnału zadanego podrzędnej pętli regulacyjnej (sygnal 'uS2' w TABELI 4).

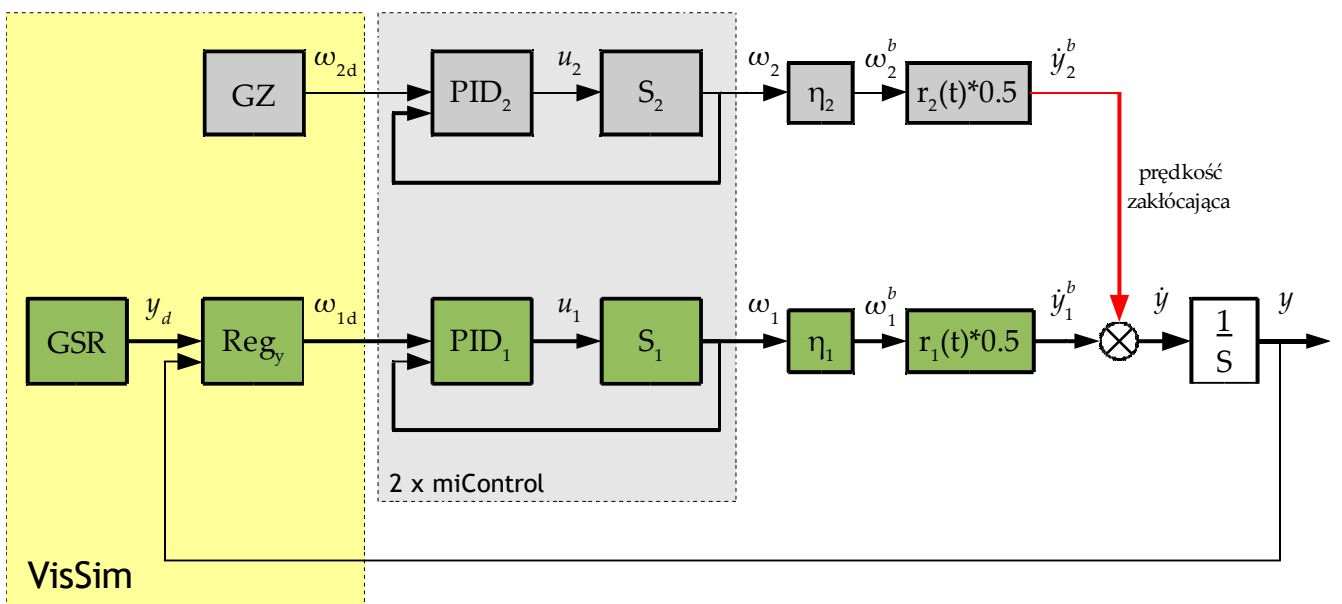
Zgodnie z zasadami teorii sterowania na stanowisku ZB2 istnieje możliwość realizacji zadań stabilizacji pozycji zwisu oraz śledzenia trajektorii pozycji zwisu z jednoczesnym tłumieniem wpływu zakłócenia (stałego lub zmiennego) oddziałującego na wyjście regulowane. Zestaw sygnałów mierzonych oraz sterujących dostępnych na stanowisku ZB2 zebrano w TABELI 4. Ogólny schemat przepływu sygnałów w systemie ZB2 przedstawia Rys. 2. Schemat blokowy układu sterowania z wyróżnionym torem zakłócenia oraz torem sterowania przedstawiono na Rys. 4.



Rys 1. Widok ogólny modelu zwijarki bębnowej ZB2



Rys 2. Schemat przepływu sygnałów w systemie ZB2 (nazwy sygnałów wg TABELI 4)



Rys 3. Schemat blokowy układu sterowania systemu ZB2

(GZ: Generator Zakłócenia, GSR: Generator Sygnału Referencyjnego, Reg_y: Regulator sygnału 'y', ω^b : prędkość kątowa bębna)

Realizacja układu sterowania i model procesu.

Układ sterowania ZB2 składa się z dwóch niezależnych torów regulacyjnych:

T1. obwodu regulacji pozycji zwisu liny (masy obciążającej) - bloki: GSR+Reg_y+PID₁+S₁

T2. obwodu realizacji programowalnego zakłócenia - bloki: GZ+PID₂+S₂

Obwody regulacji prędkości z regulatorem PID (zrealizowane z wykorzystaniem dwóch niezależnych sterowników firmy miControl) gwarantują fizyczną realizację zadanych sygnałów prędkości silników S1 i S2. W torze T1 obwód regulacji prędkości stanowi podrzędną pętlę regulacyjną, której sygnałem zadany jest wyjście bloku Reg_y (kaskadowa struktura układu regulacji). Projektowanie nadrzędnego obwodu regulacji z regulatorem Reg_y, sprzężeniem zwrotnym od pomiaru pozycji masy (sygnał 'y') oraz generatorem sygnału referencyjnego (GSR) odbywa się w całości w środowisku graficznym VisSim na komputerze PC (elementy schematu z Rys.3 zawarte w żółtym prostokącie oznaczonym komentarzem 'VisSim'). Pozostałe bloki schematu z Rys. 3 reprezentują model procesu oddziaływania napędów na zwis liny (pozycję masy obciążającej). Model ten wynika z prostej analizy kinematycznej, w której aktualna wypadkowa prędkość przemieszczania masy (zmiana pozycji zwisu w czasie) dy/dt wynika z sumy składowych prędkości dy_1^b/dt oraz dy_2^b/dt związanych z chwilowymi obwodowymi prędkościami liny dla obu bębnow zwijarki:

$$dy/dt = dy_1^b/dt + dy_2^b/dt \Rightarrow dy/dt = 0.5*(r_1 \eta_1 \omega_1) + dy_2^b/dt \quad (1)$$

gdzie $r_1=r_1(t)$ jest aktualnym promieniem walca liny nawiniętej na bęben 1 (parametr zmienny w czasie), $\eta_1 < 1$ jest współczynnikiem przełożenia przekładni redukcyjnej silnika S1, a współczynnik 0.5 wynika z faktu, iż lina rozwijana z bębna jest 'rozdzielana' na dwie części - z prawej i lewej strony przemieszczanej masy. Sygnał $dy_2^b/dt = 0.5*(r_2 \eta_2 \omega_2)$ we wzorze (1) jest traktowany jako zakłócenie procesu regulacji. Równanie (1) przedstawia zatem model obiektu całkującego ze zmiennym wzmocnieniem (obiekt niestacjonarny) z wymuszeniem sterującym ω_1 oraz zakłóceniem dy_2^b/dt . Jeżeli założymy, że w układzie z Rys. 3 zachodzi $\omega_1 \approx \omega_{1d}$ wówczas model (1) możemy zapisać jako zależny bezpośrednio od sygnału sterującego ω_{1d} dostępnego w środowisku VisSim (sygnał 'uS1' z TABELI 1):

$$\omega_1 \approx \omega_{1d} \Rightarrow dy/dt \approx 0.5*(r_1 \eta_1 \omega_{1d}) + dy_2^b/dt \quad (2)$$

Dostępne sygnały analogowe i cyfrowe w układzie sterowania ZB2 przedstawiono w TABELI 1. Dodatkowo TABELA 2 zawiera odpowiedniki nazw sygnałów dostępnych w ZB2 dla wybranych oznaczeń w układzie sterowania z Rys .3 (nie wszystkie sygnały dostępne w systemie ZB2 zostały zaznaczone na schemacie z Rys.3, a tylko te które są kluczowe dla zadań sterowania pozycją zwisu).

Opis sygnałów cyfrowych dostępnych w sterownikach miControl/Dunkermotoren zawiera TABELA 3.

TABELA 1. Zestawienie sygnałów wejściowych i wyjściowych w układzie karta I/O <-> ZB2

L.p.	Nazwa	Typ sygnału (wg ustawień karty I/O)*	Kanał karty I/O	Kierunek (względem karty I/O)	Opis
1	uS1	Analogowy ±10V	D/A OUT1	OUT (wyjście)	Sterowanie silnikiem S1 (zadana wartość prędkości S1)
2	uS2	Analogowy ±10V	D/A OUT0	OUT (wyjście)	Sterowanie silnikiem S2 (zadana wartość prędkości S2)
3	qS1	Analogowy ±10V	CH0	IN (wejście)	Pozycja silnika S1
4	vS1	Analogowy ±10V	CH1	IN (wejście)	Prędkość silnika S1
5	qS2	Analogowy ±10V	CH2	IN (wejście)	Pozycja silnika S2
6	vS2	Analogowy ±10V	CH3	IN (wejście)	Prędkość silnika S2
7	Linka	Analogowy ±10V	CH4	IN (wejście)	Bieżąca pozycja masy - sygnał regulowany (z czujnika linkowego)
----	-----	-----	-----	-----	-----
8	zeroS1	Cyfrowy	BitC0	OUT (wyjście)	Zerowanie wskazania pozycji silnika S1 (stanem zerowym)
9	zeroS2	Cyfrowy	BitC1	OUT (wyjście)	Zerowanie wskazania pozycji silnika S2 (stanem zerowym)
10	DS1	Cyfrowy	BitA0	OUT (wyjście)	Sygnał podpięty do wejścia Din1 sterownika Dunkermotoren silnika S1
11	DS2	Cyfrowy	BitA1	OUT (wyjście)	Sygnał podpięty do wejścia Din1 sterownika Dunkermotoren silnika S2
12	StartStop1	Cyfrowy	BitB0	IN (wejście)	Stan przełącznika Start/Stop 1
13	DoS1	Cyfrowy	BitB2	IN (wejście)	Stan wyjścia Dout0 sterownika Dunkermotoren silnika S1
14	DoS2	Cyfrowy	BitB3	IN (wejście)	Stan wyjścia Dout0 sterownika Dunkermotoren silnika S2
15	DIU0	Cyfrowy	BitB1	IN (wejście)	pozostawione do przyszłego wykorzystania
16	DIU1	Cyfrowy	BitB4	IN (wejście)	pozostawione do przyszłego wykorzystania
17	DIU2	Cyfrowy	BitB5	IN (wejście)	pozostawione do przyszłego wykorzystania

*Sygnały analogowe łączone z kartą I/O w trybie 'Single Ended'

TABELA 2. Zestawienie odpowiedników nazw sygnałów z TABELI 1 z oznaczeniami na Rys.3

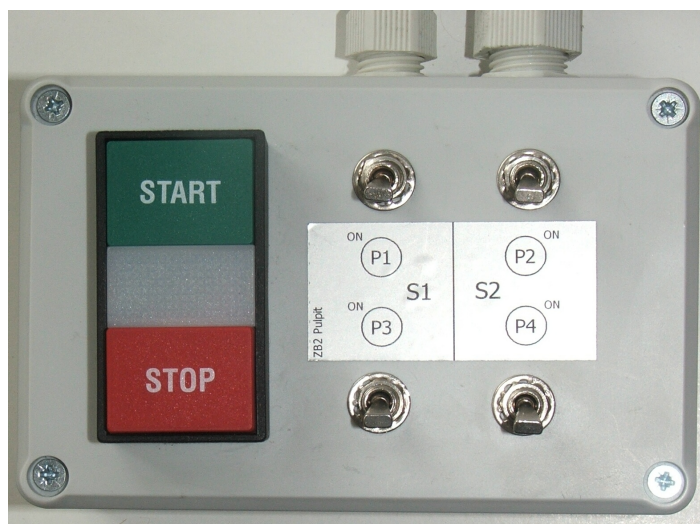
Nazwa sygnału z TABELI 1	Oznaczenie na Rys.3
uS1	ω_{1d}
uS2	ω_{2d}
vS1	ω_1
vS2	ω_2
Linka	y

TABELA 3. Zestawienie sygnałów cyfrowych sterowników Dunkermotoren BGE 3515

Silnik	Kanał	Kierunek (względem sterownika silnika)	Sygnal
S1	DIN 0	IN (wejście)	Stan przełącznika StartStop 1
S1	DIN 1	IN (wejście)	Sterowanie z karty I/O (podpięty sygnał DS1)
S1	DIN 2	IN (wejście)	Przełącznik 1 na pulpicie
S1	DIN 3	IN (wejście)	Przełącznik 3 na pulpicie
S1	DOUT 0	OUT (wyjście)	pozostawione do przyszłego wykorzystania
-----	-----	-----	-----
S2	DIN 0	IN (wejście)	Stan przełącznika StartStop 1
S2	DIN 1	IN (wejście)	Sterowanie z karty I/O (podpięty sygnał DS2)
S2	DIN 2	IN (wejście)	Przełącznik 2 na pulpicie
S2	DIN 3	IN (wejście)	Przełącznik 4 na pulpicie
S2	DOUT 0	OUT (wyjście)	pozostawione do przyszłego wykorzystania

Pulpit, wyłączniki krańcowe i zasilanie.

Rys. 4 przedstawia widok pulpitu operatora zawierający przycisk StartStop1 oraz cztery przełączniki funkcyjne - ich opis zawiera TABELA 4. Dwa wyłączniki krańcowe (górny przy czujniku linkowym oraz dolny na podstawie zwijarki) przedstawiono na Rys. 5. Opis działania i sposób zwolnienia wyłączników krańcowych zawarto w TABELI 5. Przyciska załączenia/wyłączenia zasilania modułu zwijarki ZB2 jest umieszczony na bocznej ścianie skrzynki sterowniczej zwijarki (por. Rys. 6).



Rys 4. Pulpit z przyciskiem StartStop1 oraz przełącznikami P1-P4 (przełączniki w stanie OFF)

TABELA 4. Opis funkcji przełączników na stanowisku ZB

Przełącznik	Lokalizacja	Działanie
StartStop 1	Panel	załączenie/wyłączenie wzmacniaczy mocy silników S1 i S2
P1	Panel	zatrzymanie i stabilizacja bieżącej pozycji silnika S1
P2	Panel	zatrzymanie i stabilizacja bieżącej pozycji silnika S2
P3	Panel	-- (pozostawione do przyszłego wykorzystania)
P4	Panel	-- (pozostawione do przyszłego wykorzystania)
StartStop 2	Skrzynka napędów	załączenie/wyłączenie zasilania urządzenia

TABELA 5. Opis działania wyłączników krańcowych

Wyłącznik	Działanie
Górny (linka)	Wyzwolenie wyłączza wzmacniaczy mocy silników S1 i S2; uzwojenia obu silników - zwarte; grawitacyjny zjazd masy z wyłącznika zezwala na ponowne uruchomienie wzmacniaczy. przełącznikiem 'StartStop 1'.
Dolny (podstawa)	Wyzwolenie wyłączza wzmacniaczy mocy silników S1 i S2; uzwojenia obu silników - zwarte; ponowne uruchomienie wzmacniaczy przełącznikiem 'StartStop 1' poprzez ręczne podtrzymanie stanu Start i programowy zjazd z wyłącznika (podniesienie masy silnikiem S1 lub S2).



Rys 5. Umiejscowienie wyłącznika krańcowego górnego (z lewej) oraz dolnego (z prawej)

II. WYBRANE DANE TECHNICZNE

Poniżej przedstawiono tablice z przypisaniem numerów kanałów sygnałów analogowych i cyfrowych na złączach taśmowych karty I/O typu PCI-DAS 1602/12. TABELA 6 zawiera podstawowe dane techniczne zastosowanych modułów napędowych firmy Maxon.

Table 2-2. 8-channel differential mode pin out

Signal Name	Pin			Pin	Signal Name
GND	100	■	■	50	GND
EXTERNAL INTERRUPT	99	■	■	49	SSH OUT
n/c	98	■	■	48	PC +5 V
EXTERNAL D/A PACER GATE	97	■	■	47	n/c
D/A INTERNAL PACER OUTPUT	96	■	■	46	n/c
A/D INTERNAL PACER OUTPUT	95	■	■	45	A/D EXTERNAL TRIGGER IN
n/c	94	■	■	44	D/A EXTERNAL PACER IN
n/c	93	■	■	43	ANALOG TRIGGER IN
-12 V	92	■	■	42	A/D EXTERNAL PACER
GND	91	■	■	41	CTR 4 OUT
+12 V	90	■	■	40	CTR 4 GATE
GND	89	■	■	39	CTR 4 CLK
n/c	88	■	■	38	D/A OUT 1
n/c	87	■	■	37	D/A GND 1
n/c	86	■	■	36	D/A OUT 0
n/c	85	■	■	35	D/A GND 0
n/c	84	■	■	34	n/c
n/c	83	■	■	33	n/c
n/c	82	■	■	32	n/c
n/c	81	■	■	31	n/c
n/c	80	■	■	30	n/c
n/c	79	■	■	29	n/c
n/c	78	■	■	28	n/c
n/c	77	■	■	27	n/c
n/c	76	■	■	26	n/c
n/c	75	■	■	25	n/c
FIRSTPORTC Bit 7	74	■	■	24	n/c
FIRSTPORTC Bit 6	73	■	■	23	n/c
FIRSTPORTC Bit 5	72	■	■	22	n/c
FIRSTPORTC Bit 4	71	■	■	21	n/c
FIRSTPORTC Bit 3	70	■	■	20	n/c
FIRSTPORTC Bit 2	69	■	■	19	n/c
FIRSTPORTC Bit 1	68	■	■	18	LLGND
FIRSTPORTC Bit 0	67	■	■	17	CH7 LO
FIRSTPORTB Bit 7	66	■	■	16	CH7 HI
FIRSTPORTB Bit 6	65	■	■	15	CH6 LO
FIRSTPORTB Bit 5	64	■	■	14	CH6 HI
FIRSTPORTB Bit 4	63	■	■	13	CH5 LO
FIRSTPORTB Bit 3	62	■	■	12	CH5 HI
FIRSTPORTB Bit 2	61	■	■	11	CH4 LO
FIRSTPORTB Bit 1	60	■	■	10	CH4 HI
FIRSTPORTB Bit 0	59	■	■	9	CH3 LO
FIRSTPORTA Bit 7	58	■	■	8	CH3 HI
FIRSTPORTA Bit 6	57	■	■	7	CH2 LO
FIRSTPORTA Bit 5	56	■	■	6	CH2 HI
FIRSTPORTA Bit 4	55	■	■	5	CH1 LO
FIRSTPORTA Bit 3	54	■	■	4	CH1 HI
FIRSTPORTA Bit 2	53	■	■	3	CH0 LO
FIRSTPORTA Bit 1	52	■	■	2	CH0 HI
FIRSTPORTA Bit 0	51	■	■	1	LLGND
PCI slot ↓					

Table 2-3. 16-channel single-ended mode pin out

Signal Name	Pin			Pin	Signal Name
GND	100	■	■	50	GND
EXTERNAL INTERRUPT	99	■	■	49	SSH OUT
n/c	98	■	■	48	PC +5 V
EXTERNAL D/A PACER GATE	97	■	■	47	n/c
D/A INTERNAL PACER OUTPUT	96	■	■	46	n/c
A/D INTERNAL PACER OUTPUT	95	■	■	45	A/D EXTERNAL TRIGGER IN
n/c	94	■	■	44	D/A EXTERNAL PACER IN
n/c	93	■	■	43	ANAL OG TRIGGER IN
-12 V	92	■	■	42	A/D EXTERNAL PACER
GND	91	■	■	41	CTR 4 OUT
+12 V	90	■	■	40	CTR 4 GATE
GND	89	■	■	39	CTR 4 CLK
n/c	88	■	■	38	D/A OUT 1
n/c	87	■	■	37	D/A GND 1
n/c	86	■	■	36	D/A OUT 0
n/c	85	■	■	35	D/A GND 0
n/c	84	■	■	34	n/c
n/c	83	■	■	33	n/c
n/c	82	■	■	32	n/c
n/c	81	■	■	31	n/c
n/c	80	■	■	30	n/c
n/c	79	■	■	29	n/c
n/c	78	■	■	28	n/c
n/c	77	■	■	27	n/c
n/c	76	■	■	26	n/c
n/c	75	■	■	25	n/c
FIRSTPORTC Bit 7	74	■	■	24	n/c
FIRSTPORTC Bit 6	73	■	■	23	n/c
FIRSTPORTC Bit 5	72	■	■	22	n/c
FIRSTPORTC Bit 4	71	■	■	21	n/c
FIRSTPORTC Bit 3	70	■	■	20	n/c
FIRSTPORTC Bit 2	69	■	■	19	n/c
FIRSTPORTC Bit 1	68	■	■	18	LLGND
FIRSTPORTC Bit 0	67	■	■	17	CH15 HI
FIRSTPORTB Bit 7	66	■	■	16	CH7 HI
FIRSTPORTB Bit 6	65	■	■	15	CH14 HI
FIRSTPORTB Bit 5	64	■	■	14	CH6 HI
FIRSTPORTB Bit 4	63	■	■	13	CH13 HI
FIRSTPORTB Bit 3	62	■	■	12	CH5 HI
FIRSTPORTB Bit 2	61	■	■	11	CH12 HI
FIRSTPORTB Bit 1	60	■	■	10	CH4 HI
FIRSTPORTB Bit 0	59	■	■	9	CH11 HI
FIRSTPORTA Bit 7	58	■	■	8	CH3 HI
FIRSTPORTA Bit 6	57	■	■	7	CH10 HI
FIRSTPORTA Bit 5	56	■	■	6	CH2 HI
FIRSTPORTA Bit 4	55	■	■	5	CH9 HI
FIRSTPORTA Bit 3	54	■	■	4	CH1 HI
FIRSTPORTA Bit 2	53	■	■	3	CH8 HI
FIRSTPORTA Bit 1	52	■	■	2	CH0 HI
FIRSTPORTA Bit 0	51	■	■	1	LLGND

PCI slot ↓

TABELA 6. Dane techniczne silników napędowych zwijarki

Silnik	Typ (Maxon)	Czujnik (Maxon)	Przekładnia	Uwagi
S1	A-max 32 szczotkowy DC (236668)	obr.-impulsowy: 4*1000 imp/obr MR typ L (228456)	1:21 (GP 32A: 166160)	Napęd lewostronnego bębna ZB2 (patrząc od frontu); funkcja - sterowanie położeniem masy
S2	RE 30 szczotkowy DC (268193)	obr.-impulsowy: 4*1000 imp/obr MR typ L (228456)	1:51 (GP 32C: 166939)	Napęd prawostronnego bębna ZB2 (patrząc od frontu); funkcja - generacja zakłócenia położenia masy

III. OPROGRAMOWANIE

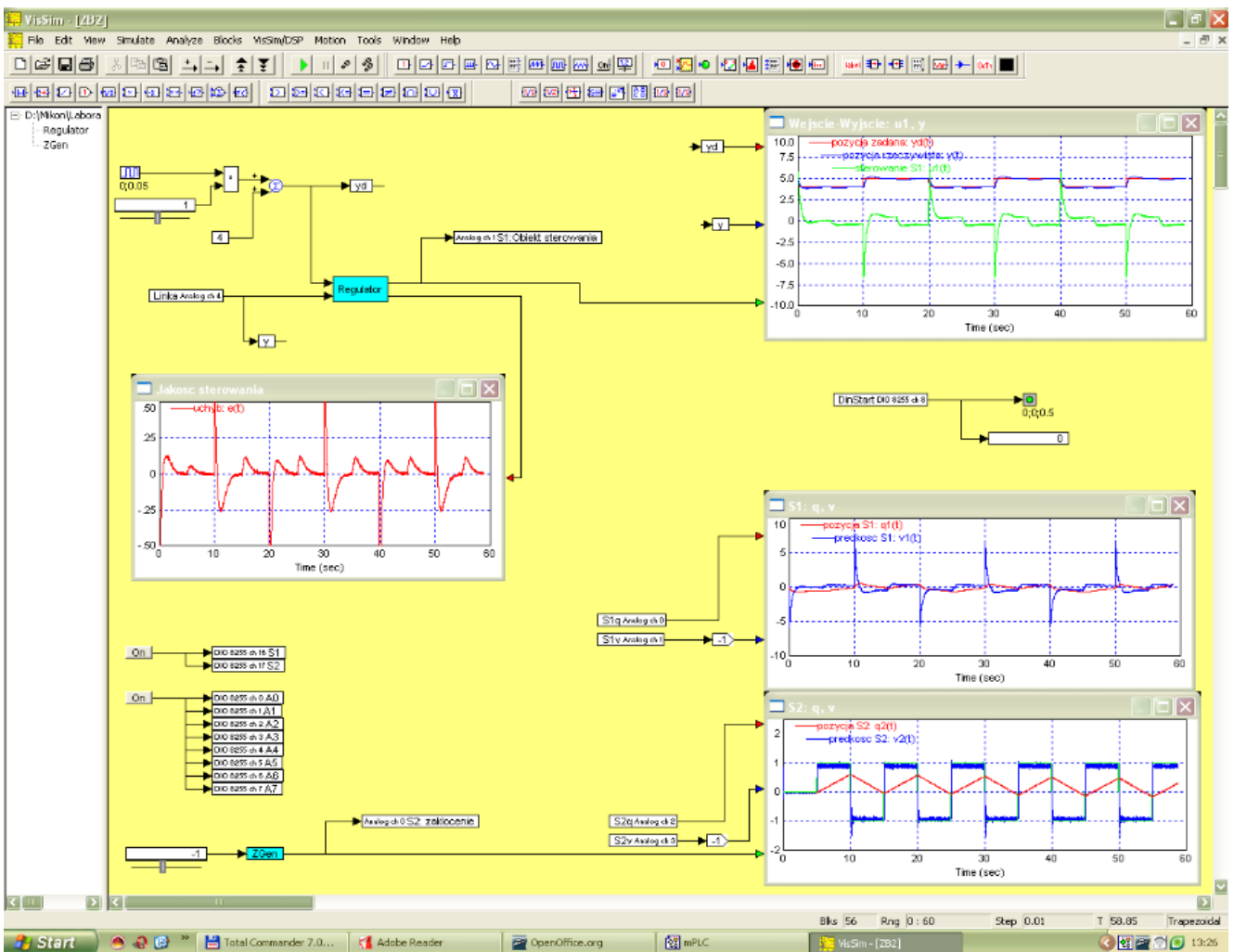
W skład programów koniecznych do uruchomienia stanowiska ZB2 wchodzi plik z programem sterującym modułami miControl/Dunkermotoren BGE 3515 (listing programu zawarto poniżej) oraz plik VisSim-a zawierający podstawowy schemat blokowy pozwalający na sterowanie modułem zwijarki (Rys. 7).

Zbiór plików do stanowiska ZB2:

- **ZB2.vsm** - podstawowy schemat środowiska VisSim ilustrujący działanie ZB2
- **ZB2AinVelocity.py** - skrypt środowiska Python z programem sterowników Dunkermotoren/miControl ładowany poprzez magistralę CAN (gniazdo DB9 z prawej strony obudowy skrzynki napędów - Rys. 6)



Rys 6. Skrzynka sterownicza ZB2 (widok z boku)



Rys 7. Schemat blokowy podstawowego układu sterowania systemem ZB2 w środowisku VisSim

Listing programu sterowników miControl/Dunkermotoren BGE 3515:

```

****START*****
from mc.dsa import *
from mc.mpu import *          # Motion Process Unit

m = None

#- Application parameter -----
def ControllerParam (d):

    # tutaj moze Pan wpisac parametry konfigurujace sterownik

    d.Disable()                # disable power stage
    d.ClrError()               # erase potential errors
    d.DefaultParam()           # set default parameter
    d.MotorTypeDC()            # Motortype      = DC
    d.VelFeedbackEnc()         # Refeeding     = Encoder
    d.MotorEncResolution(4000) # encoder resolution = 4000

    # MotorPolarity - Conformity of the direction of speed and direction of counts
    # of the hall sensors. Please adjust d.MotorPolarity(1) if the control circuit is uncontrolled
    
```

(the engine shaft rotates at full speed).

```
d.FctControl(3)          # activates Fct parameter (Factorgroup)
                        # It converts the position units in [rev]
                        # (in revolutions of the engine shaft)

d.MotorPolarity(1)
d.PwmMode(1)            #PWM unipolarny!

d.ModeVel()             # Operating mode = velocity
d.VelSourceAin(0)       # predkosc zadana - z wejscia analogowego Ain0
d.VelReference(4000)    # Scale: 10V on AIN0 = 4000[rpm]
d.VelLimitMaxPos(4000)
d.VelLimitMaxNeg(4000)

d.MesVelTime(1)         #okres pomiaru predkosci silnika Tv = 1ms
d.VelSampleTime(1000)  #okres próbkowania petli regulacyjnej: Tp = 1ms

d.VelAcc_dV(1000)       # acceleration = 1000[rpm]/1[ms]
d.VelAcc_dT(1)
d.VelDec_dV(1000)       # deceleration = 1000[rpm]/1[ms]
d.VelDec_dT(1)
d.VelDecQuickStop_dV(4000)
d.VelDecQuickStop_dT(1)
d.VelRampEnable()      # activate the ramp generator

d.VelKp(80)
d.VelKi(10)
d.VelKd(10)
d.VelKvff(0)           #wspolczynnik sprzezenia wyprzedzajacego (feed-forward)
d.VelILimit(1000)      #ograniczenie wartosci calki do 1000 [mV]
d.DevDinStartStop(2, active_level= -1) #Din2 jako start/stop (aktywny stan: 0)

d.Enable()

return
```

```
#- MPU-Program -----
def MpuProgram (m):
    m.Jmp("Start")

    # MPU Variables
    #
    m.Var0 = [m.PC(), 0]          # 8-Bit          index = 0x5101, subindex = 0
    m.Calibrate = [m.PC(), 1]     # 32-Bit          1
    m.Debug = [m.PC(), 2]        # 32-Bit          2
    m.End()

    m.Var1 = [m.PC(), 0]          # 8-bit           index = 0x5102, subindex = 0
    m.Var_Refer = [m.PC(), 1]     # 32-bit          1
    m.Var_Calib = [m.PC(), 2]     # 32-bit          2
    m.End()

    #powyzsze wartosci mozna odczytywac z pozycji pythona d.SdoRd(0x5101,1), oraz zapisywac d.SdoWr(0x5101,1,1)

    # MPU Start
    #
    m.Label("Start")
    m.Mov([0x0004,0],1)          #zalaczenie wzmacniacza mocy (zapis do komorki 3004.00 wartosci 1)

    m.Label("MainLoop")         # MPU-Main Loop

    m.Mov(Accu,m.Calibrate)
    m.Jeq("NotCalibrate",0)

    #*****
    m.Mov(Reg5,0)                #inicjalizacja Reg5
    m.Mov(Reg6,1024)             #licznik petli
```

```

m.Label("Measuring")           #petla pomiarowa
m.Madd(Reg5,[0x0100,0])        #Reg5:=Reg5+Ain0
m.Dec(Reg6)
m.Mov(Accu,Reg6)
m.Jneq("Measuring",0)
m.Mshr(Reg5,10)                #koniec petli - usrednianie wyniku pomiaru
m.Mov(m.Var_Calib,Reg5)       #zaladowanie wyniku

#m.Mov(m.Var_Calib,[0x0100,0]) # Read analog input Ain0   analogicznie do 0x3100 z listy parametrow
m.Mov(m.Calibrate,0)          #wskazanie, ze pomiary wykonane
#*****

m.Label("NotCalibrate")

# Calculate limits
m.Mov(Reg2,m.Var_Calib)       # m.LimitMin
m.Mov(Reg3,10000)             # m.LimitMax

m.Mov(Accu,10000)
m.Sub(Accu,m.Var_Calib)
m.Msub(Reg2,Accu)
m.Mov(m.Debug,Reg2)

# Calc. Velocity
m.Mov(Accu,[0x0100,0])        # Read analog input Ain0
m.LimitMini_i32(Accu,Reg2)
m.LimitMaxi_i32(Accu,Reg3)
m.Sub(Accu,m.Var_Calib)
m.Mul(Accu,m.Var_Refer)
m.Mov(Reg1,10000)
m.Msub(Reg1,m.Var_Calib)
m.Div(Accu,Reg1)              # wynik: Vel [rpm] = (AIN - Calibration)*Reference/(10000 - Calibration)

m.Mov([0x500,0],Accu)         # Set target speed SVel
m.Jmp("MainLoop")           # Jump to main loop label

#- Project helper function -----
#
# save_mpu True : MPU Programm will be stored
# False: MPU Programm will not be stored
#
def mcDSA_C5_Vel_Mode_5V_AIN (node_id, dump_mpu=False, save_mpu=False, start_mpu=False):
    global m

    d = Dsa(node_id)           # Dsa-Instance
    m = Mpu(d)                 # Mpu-Instance

    m.Break()

    ControllerParam(d)
    MpuProgram(m)              # Build the MPU-Programm

    #test!!!
    print "VelKp =", d.VelKp()
    print "VelKi =", d.VelKi()
    print "VelKd =", d.VelKd()
    print "Offset_Low = ", Reg5

    #koniec testu!!!

    if dump_mpu:
        m.Prg.Dump()

    if save_mpu or start_mpu:
        print "Firmware-Version =", d.VersionSwString()

```

```
d.Disable()
m.Prg.Break()          # MPU-Program break
m.Prg.Load()          # Load the MPU program in the device

if save_mpu:
    print "Programming device", node_id, "..."

    try:
        d.Disable()      # StoreMpu works only in Disable state
        d.StoreParam()  # Stores Parameters to the EEPROM
        m.Prg.AutoStart(1) # Set the AutoStart
        m.Prg.Store()   # Store MPU-Programm and Settings in EEPROM

        print "Done."
    except:
        print "ERROR: Device not programmed."

if start_mpu:
    m.Prg.Start()      # Start loaded MPU program (skok do etykiety "Start")

del d
del m

#-----
if __name__ == "__main__":
    mcDSA_C5_Vel_Mode_5V_AIN (1, dump_mpu=True,save_mpu=True,start_mpu=True)    # node id sterownika: 1=S1(Amax),
    2=S2(RE30)

****KONIEC****
```

Do niniejszego dokumentu dołączono także:

1. schematy obwodów elektrycznych modułu zwijarki
2. rysunki techniczne elementów mechaniki modułu zwijarki (pliki AutoCAD)
3. karty katalogowe wybranych podzespołów i części zwijarki
4. instrukcję do karty PCI-DAS 1602/12 (plik *.pdf)
5. instrukcje do oprogramowania VisSim oraz nakładki RealTimePRO (pliki *.pdf)

Aktualizacja:
7.01.2010

opracowanie:
Maciej Michatek