ZASTOSOWANIE WYNIKÓW ANALIZY OBRAZÓW DO KOREKTY UKŁADU ODNIESIENIA OBIEKTU W ZROBOTYZOWANYM GNIEŹDZIE OBRÓBCZYM

Andrzej Burghardt[1a], Krzysztof Kurc[1b], Dariusz Szybicki[1c], Paweł Obal[1d]

[1]Katedra Mechaniki Stosowanej i Robotyki, Politechnika Rzeszowska

[a]andrzejb@prz.edu.pl, [b]kkurc@prz.edu.pl, [c]dszybicki@prz.edu.pl, [d]p.obal@prz.edu.pl

Streszczenie

W ostatnich latach silnie poszerza się spektrum wykorzystania robotów w ślusarskich w procesach obróbczych. Zastosowanie robotów wymaga powtarzalności geometrii i zdefiniowania pozycji i orientacji obrabianych detali. Jeżeli nie ma możliwości zapewniania wystarczającej powtarzalności lub programowanie skomplikowanych trajektorii przeprowadzane jest metodą offline, koniecznie jest zastosowanie układów korekcji ścieżki, takich jak np. system wizyjny. W pracy przedstawiono stanowisko składające się z systemu wizyjnego, pozycjonera oraz robota przemysłowego. Wygenerowana trajektoria robota w układzie odniesienia obiektu może być przemieszczana zgodnie z wektorem translacji i rotacji otrzymywanym na podstawie pomiaru dokonywanego z zastosowaniem systemu wizyjnego. Działanie zaproponowanego rozwiązania zostało zweryfikowane na obiekcie rzeczywistym.

Słowa kluczowe: robotyka przemysłowa, system wizyjny, adaptacja ścieżki robota

APPLICATION OF IMAGE ANALYZE RESULTS TO CORRECTION OF THE OBJECT COORDINATE SYSTEM IN THE ROBOTIC SYSTEMS

Summary

In recent years there has been an increase of the use of robots for locksmith technological processes. It requires repeatability of the geometry and proper settings of a workpiece. If it is not possible to provide sufficient repeatability or programming of complicated trajectories is performed offline, it is essential to use path correction systems, such as the vision system. In the article a station with a vision system, positioner and industrial robot was presented. The generated trajectory of the robot in the object reference system can be moved according to the translation and rotation vector obtained from the measurement of the vision system. Received solution has been verified on a real object.

Keywords: industry robotics, vision system, robot trajectory adaptation

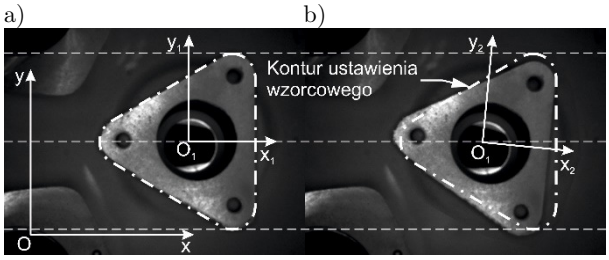
1. WSTĘP

Genezą problemu jest próba zrobotyzowania procesu gratowania elementów silników lotniczych, w których występują części o kształcie geometrycznym losowo zmiennym w ograniczonym zakresie [2,4]. Wynika to ze stosowanej technologii odlewania precyzyjnego, w której dokładność kształtu zależy od precyzji wykonania form

oraz występowania skurczu w czasie krzepnięcia. Brak wystarczającej powtarzalności geometrii obrabianych elementów uniemożliwia zastosowanie robotów do wykonywania procesów obróbczych bez użycia układów kompensujących przesunięcie ścieżki [9]. Aktualnie większość czynności technologicznych związanych z wytwarzaniem tego typu elementów jest wykonywana ręcznie, co generuje wysokie koszty wytwarzania oraz zwiększa ryzyko powstania elementów brakowych, spowodowanych błędem człowieka. W takich przypadkach wymagane jest zastosowanie układów korekcji ścieżki punktu TCP (z ang. tool center point) robota. Najczęściej stosowane rozwiązania adaptacji trajektorii punktu TCP to [1,3,5,6]:

- narzędzia aktywne:
  - narzędzia pneumatyczne o regulowanej sile docisku,
  - manipulatory z systemem kontroli siły,
- systemy optyczne:
  - laserowe systemy adaptacji trajektorii,
  - systemy wizyjne,
- aplikacje uczące,
- narzędzia dedykowane.

W niniejszej pracy przedstawiono propozycję rozwiązania problemu adaptacji trajektorii punktu TCP poprzez zastosowanie systemu wizyjnego. Korekcja dokonywana jest na podstawie przeprowadzonej analizy pozycji i orientacji obrabianej krawędzi względem elementu wzorcowego. Prace badawcze nad rozwiązaniem problemu przeprowadzono dla procesu zatępiania krawędzi dyfuzora silnika V2500. Na rys. 1 przedstawiono porównanie ustawienia dwóch elementów dyfuzora, przeznaczonych do obróbki.



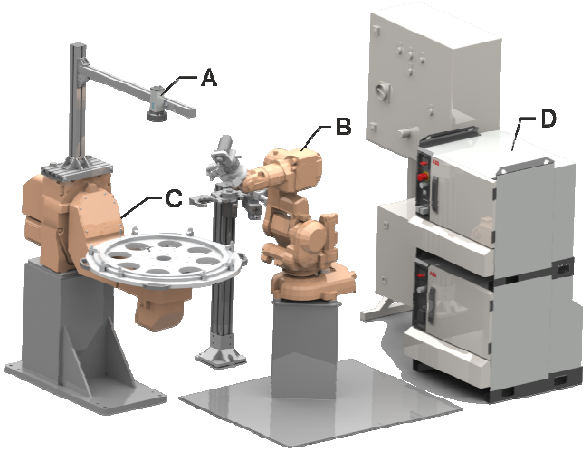
Rys. 1. Porównanie pozycji elementu: a) wzorcowego, b) losowej próbki

Element wzorcowy został wykorzystany do zaprogramowania ścieżki robota. Element obrabiany może się przemieszczać w sposób pokazany na rys. 1. Wynika to z technologii wytwarzania. Korpus silnika jest odlewem i elementy typu naba podczas krzepnięcia mogą zmieniać pozycję i orientacje w płaszczyźnie xy, w ograniczonym zakresie (maksymalna obserwowana zmiana pozycji naby wynosi ±2 [mm], a maksymalna obserwowana rotacja wynosi ±0,06 [rad]). Co za tym idzie, ścieżka punktu TCP robota zaprogramowana dla wzorca musi być skorygowana w celu wykonania procesu obróbki każdego elementu osobno.

2. STANOWISKO BADAWCZE

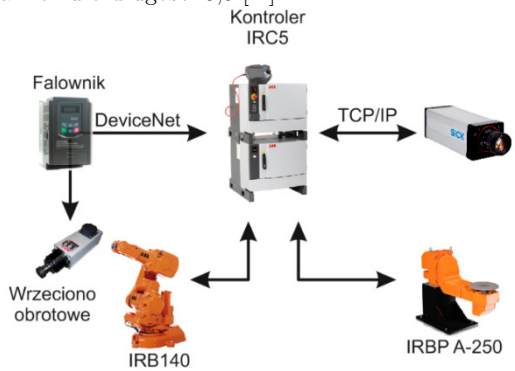
W celu przeprowadzenia badań wizyjnej adaptacji układu odniesienia zaproponowano stanowisko zrobotyzowane wyposażone w:

A. kamerę systemu wizyjnego SICK IVC-2DM1131 z modułem oświetleniowym SICK ICL300-F202S01,
B. robot ABB IRB140,
C. pozycjoner ABB A250,
D. kontroler ABB IRC5.



Rys. 2. Stanowisko obróbcze

Robot B (rys. 2) wykonuje operacje ślusarskie przy użyciu narzędzi skrawających. Element obrabiany jest mocowany na pozycjonerze C (rys. 2), co umożliwia automatyczną zmianę ustawienia elementu względem robota i kamery. Detal mocowany jest na pozycjonerze pasownie i dodatkowo zawiera miejsca na umieszczenie kołków pozycjonujących, wykonanych w tolerancji H7/h7. Kamera A (rys. 2) jest zamontowana nad detalem i przymocowana do konstrukcji pozycjonera. Pozycjoner ustawia detal przed obiektywem kamery, aby umożliwić proces akwizycji obrazu obrabianych elementów. Rozdzielczość pomiarowa kąta obrotu pozycjonera wynosi ±0,0001 [rad], producent podaje także wartość powtarzalności pozycjonowania ±0,05 [mm] zdefiniowana na ramieniu o długości 0,5 [m].



Rys. 3. Schemat komunikacji urządzeń stanowiska zrobotyzowanego

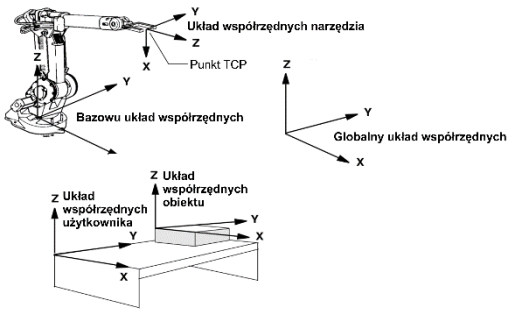
Dokładność realizacji ruchu przez pozycjoner nie wpływa mierzalnie przez system wizyjny na zmianę przemieszczenia i rotacji naby. Akwizycja obrazu jest wyzwalana przez kontroler robota IRC5 (rys. 2). Na rys. 3 przedstawiono schemat komunikacji poszczególnych urządzeń stanowiska zrobotyzowanego. Zastosowany system wizyjny komunikuje się z kontrolerem robota z wykorzystaniem protokołu TCP/IP. Do sterowania pracą wrzeciona zastosowano przemiennik częstotliwości, podłączony do kontrolera z zastosowaniem protokołu DeviceNet. Kontroler IRC5 steruje również ruchem robota i pozycjonera. Został wyposażony w dodatek MultiMove, pozwalający na wykonywanie przez kilka jednostek mechanicznych skoordynowanych ruchów. Dodatkowo kontroler został wyposażony w opcje Absolute Accuracy, która zwiększa dokładność pozycjonowania poprzez dodanie do oprogramowania kontrolera tablicy poprawek kompensujących błędy pozycjonowania punktu TCP robota. Opcja Absolute Accuracy jest wymagana przy stosowaniu metody adaptacji układu odniesienia z wykorzystaniem systemu wizyjnego.

3. PROCES ADAPTACJI

Zaproponowana metoda polega na zdefiniowaniu układu współrzędnych obiektu i utworzeniu w tym układzie ścieżki dla narzędzia robota. Celem zastosowania systemu wizyjnego jest porównanie, na podstawie wyników analizy obrazów, ustawienia elementów obrabianych względem ustawienia elementu wzorcowego, dla którego określono układ współrzędnych obiektu oraz ścieżkę narzędzia robota. Następnie zebrane informacje z systemu wizyjnego w postaci wartości przekształceń układu współrzędnych obiektu są przekazywane do kontrolera robota, który umożliwia korekcję układu współrzędnych obiektu. Korekta układu odniesienia obiektu powoduje także korektę ustawienia zaprogramowanej w tym układzie ścieżki.

3.1 UKŁAD WSPÓŁRZĘDNYCH OBIEKTU

Do programowania manipulatorów firmy ABB wykorzystuje się język Rapid, który pozwala na definiowanie przez użytkownika własnych układów odniesienia, których liczba nie jest ograniczona. Polega to na utworzeniu w programie zmiennej typu *wobjdata*, znajdują się w niej informacje o położeniu i orientacji zdefiniowanych układów współrzędnych użytkownika oraz obiektu (rys. 4). Układ współrzędnych użytkownika, *uframe* jest definiowany dla obrabianej powierzchni lub zamocowania detalu względem globalnego układu odniesienia. Układ współrzędnych obiektu *oframe* jest definiowany dla obrabianego przedmiotu względem układu współrzędnych użytkownika.



Rys. 4. Układy odniesienia użytkownika oraz obiektu w danej typu *wobjdata*

Adaptacja układu odniesienia jest możliwa poprzez zmianę parametrów zmiennej typu *wobjdata*. W momencie zdefiniowania zmiennej, układy *uframe* oraz *oframe* pokrywają się, tzn. współrzędne układu odniesienia obiektu są zerowe. Wykorzystując instrukcje języka Rapid, można modyfikować położenie i orientacje układu odniesienia obiektu względem układu użytkownika. W dalszej części przedstawiono fragment programu do obróbki krawędzi elementu silnika, na początku którego przesunięto położenie układu odniesienia obiektu o 2 mm względem osi x.

Listing 1.

```
PROC P_Naba_tr_cut()
  W_Naba_tr_cut.oframe.trans.x := 2;! przesunięcie układu obiektu o 2 mm
  MoveL T_Nab_tr_c_10,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveL T_Nab_tr_c_20,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveC
T_Nab_tr_c_30,T_Nab_tr_c_40,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveL T_Nab_tr_c_50,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveC
T_Nab_tr_c_60,T_Nab_tr_c_70,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveL T_Nab_tr_c_80,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
  MoveC
T_Nab_tr_c_90,T_Nab_tr_c_10,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;
ENDPROC
```

Translację oraz rotacje względem pozostałych osi przeprowadza się analogicznie, wprowadzając dane do określonych parametrów zmiennej typu *wobjdata.*

3.2 SYSTEM WIZYJNY

Zadaniem systemu wizyjnego jest określenie położenia obrabianego detalu. System umożliwia akwizycję obrazu z kamery, określenie pozycji i orientacji elementu względem wzorca, następnie przesyła wyznaczone wartości rotacji i translacji układu odniesienia obiektu do kontrolera IRC 5. Przesłane dane służą do adaptacji ścieżki robota, zapisanej w układzie odniesienia obiektu. Rozdzielczość kamery systemu wizyjnego to 1600x1200 [pix], która jest ograniczona programowo do rozmiarów 1440x1151 [pix] (rys. 6). Rozmiar pozyskanego obrazu to 108x86 [mm]. Przy tak przyjętych założeniach otrzymuje się rozdzielczość pomiarową rzędu 0,075 [mm]. Wyzna-

czona dokładność została określona w procesie kalibracji systemu wizyjnego. Elementy zamocowano na pozycjonerze, który ustawia je w zaprogramowanych pozycjach, pozwalających na dokonanie pomiaru dla wszystkich elementów korpusu silnika w jednakowym ustawieniu względem kamery. Do obsługi systemu wizyjnego firmy SICK służy oprogramowanie IVC Studio [7, 10]. Działanie systemu determinuje program pisany w formie zestawu instrukcji wykonywanych krok po kroku, przedstawionych graficznie jako tabele parametrów. W opisanym rozwiązaniu zaproponowano program podzielony na 5 podprogramów, tj.:

- akwizycja obrazu,
- przetwarzanie obrazu,
- wyznaczenie rotacji,
- wyznaczenie translacji,
- przesłanie danych.

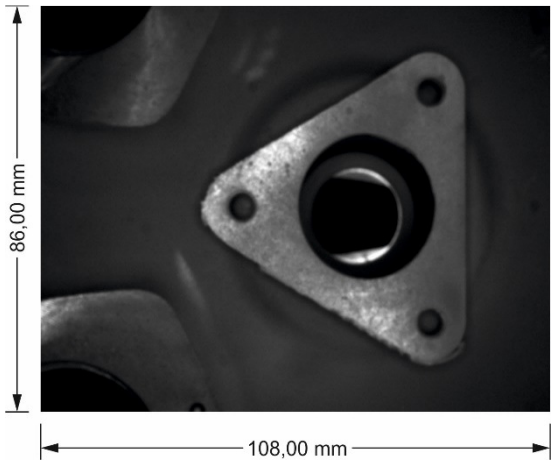
Napisany program przedstawiono poniżej w formie obrazów z interfejsu programu IVC Studio. W pierwszej części programu (rys. 5) kamera wykonuje zdjęcie analizowanego obiektu. Wyzwalaczem uruchamiającym program jest zmiana sygnału cyfrowego na wejściu systemu wizyjnego ze stanu niskiego na wysoki. Działanie wejść cyfrowych definiowane jest w programie przy pomocy instrukcji *Read Input*.

Pobieranie obrazu

0	Subroutine Start					
	01 = END step	5	-----	-----	-----	
	Time of execution (µs)	10				
	01 = Calling step	0				
1	Grab Setup					
	01 = Exposure time (µs)	500	-----	-----	-----	
	02 = Gain	450	-----	-----	-----	
	03 = External trigger	False	-----	-----	-----	
	04 = Trigger slope	Falling	-----	-----	-----	
	05 = Strobe enable	True	-----	-----	-----	
	06 = Strobe time (µs)	1000	-----	-----	-----	
	Time of execution (µs)	0				
2	Read Input [Not emulated]					
	01 = Input number	0	-----	-----	-----	
	02 = Wait	True	-----	-----	-----	
	03 = Wait for	High	-----	-----	-----	
	Time of execution (µs)	0				
	01 = Signal value	0				
3	Grab					
	01 = Destination bank	0	-----	-----	-----	
	Time of execution (µs)	15321				
	01 = File Name	'polozenie_1.bmp				
4	Goto					
	01 = Goto step	6	-----	-----	-----	
	Time of execution (µs)	1				
5	End					
	01 = Start step	0	-----	-----	-----	
	Time of execution (µs)	0				

Rys. 5. Krok 0-5 – akwizycja obrazu

Sygnał wyzwalany jest z kontrolera robota, w jego oprogramowaniu zdefiniowano przy jakim ustawieniu elementu obserwowanego należy wykonać zdjęcie. Wynikiem działania tego podprogramu jest obraz przedstawiony na rys. 6, który jest przekazany do dalszej obróbki.



Rys. 6. Pobrany obraz detalu

Kolejnym etapem działania algorytmu jest przetwarzanie pobranego obrazu w celu uzyskania potrzebnych informacji o obiekcie. W przedstawionym programie (rys. 7) wykorzystano instrukcję Blob Finder.

Wyszukiwanie obszarow Blob

6	Subroutine Start					
	01 = END step	12	-----	-----	-----	
	Time of execution (µs)	0				
	01 = Calling step	4				
7	ROI Rectangle					
	01 = X offset	0	-----	-----	-----	
	02 = Y offset	0	-----	-----	-----	
	03 = X coordinate	122	-----	-----	-----	
	04 = Y coordinate	0	-----	-----	-----	
	05 = Width	1440	-----	-----	-----	
	06 = Height	1151	-----	-----	-----	
	Time of execution (µs)	0				
	Wyszukiwanie środka dużego otworu					
8	Blob Finder					
	01 = Source Bank	0	-----	-----	-----	
	02 = ROI Definition Step	7	-----	-----	-----	
	03 = Destination Bank	1	-----	-----	-----	
	04 = Generate Blob Image	Enabled	-----	-----	-----	
	05 = Lower Threshold	0	-----	-----	-----	
	06 = Upper Threshold	78	-----	-----	-----	
	07 = Threshold Mode	Manual	-----	-----	-----	
	08 = Min Blob Area (pixels)	145000	-----	-----	-----	
	09 = Max Blob Area (pixels)	155000	-----	-----	-----	
	10 = Timeout (ms)	1000	-----	-----	-----	
	11 = Discard Edge Blobs	Disabled	-----	-----	-----	
	12 = Fill Holes in Blobs	Enabled	-----	-----	-----	
	13 = Sort By	None	-----	-----	-----	
	14 = Sorting Order	Ascending	-----	-----	-----	
	15 = Sorting Point X	0	-----	-----	-----	
	16 = Sorting Point Y	0	-----	-----	-----	
	17 = Table Index	50	-----	-----	-----	
	18 = Max Blobs in Table	1	-----	-----	-----	
	Time of execution (µs)	0				
	01 = Number of Found Blobs	0				
	02 = Lower Threshold Used	0				
Setup	03 = Upper Threshold Used	38				

Rys. 7. Krok 7-8 – wyszukiwanie środka ciężkości dużego otworu

Instrukcja *Blob Finder* przekształca obraz monochromatyczny na obraz binarny oraz określa położenie środków ciężkości obszarów uzyskanych w wyniku binaryzacji obrazu. Instrukcja *Blob Finder* jest wywoływana trzy razy, dla trzech różnych parametrów binaryzacji obrazu, aby obliczyć środki ciężkości 3 różnych obszarów naby. Program pokazany na rys. 7 przetwarza pobrany obraz (rys. 6), wyszukując punkt środkowy dużego otworu naby [8], na podstawie obrazu pokazanego na rys. 10a.

	Blob Finder				
	01 = Source Bank	0	.....	.....	.....
	02 = ROI Definition Step	7	.....	.....	.....
	03 = Destination Bank	2	.....	.....	.....
	04 = Generate Blob Image	Enabled	.....	.....	.....
	05 = Lower Threshold	60	.....	.....	.....
	06 = Upper Threshold	110	.....	.....	.....
	07 = Threshold Mode	Manual	.....	.....	.....
	08 = Min Blob Area (pixels)	470000	.....	.....	.....
	09 = Max Blob Area (pixels)	520000	.....	.....	.....
9	10 = Timeout (ms)	1000	.....	.....	.....
	11 = Discard Edge Blobs	Disabled	.....	.....	.....
	12 = Fill Holes in Blobs	Enabled	.....	.....	.....
	13 = Sort By	None	.....	.....	.....
	14 = Sorting Order	Ascending	.....	.....	.....
	15 = Sorting Point X	0	.....	.....	.....
	16 = Sorting Point Y	0	.....	.....	.....
	17 = Table Index	55	.....	.....	.....
	18 = Max Blobs in Table	1	.....	.....	.....
	Time of execution (µs)	0			
	01 = Number of Found Blobs	0			
	02 = Lower Threshold Used	60			
Setup	03 = Upper Threshold Used	100			

Rys. 8. Krok 9 – wyszukiwanie środka ciężkości kształtu naby

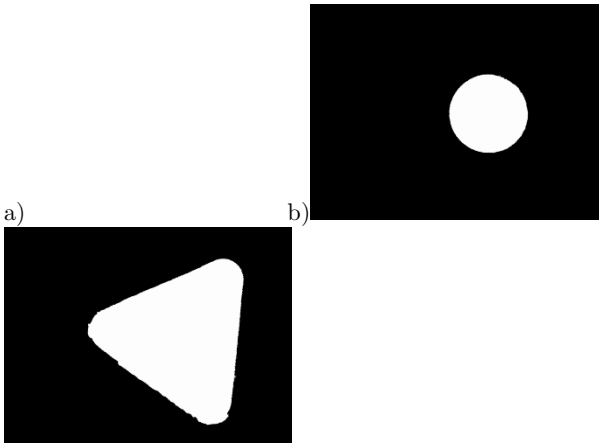
Krok dziewiąty programu, pokazany na rys. 8, umożliwia wyszukanie środka ciężkości kształtu naby na postawie obrazu pokazanego na rys. 10b.

Wyszukiwanie 3 środków małych otworów					
	Blob Finder				
	01 = Source Bank	0	.....	.....	.....
	02 = ROI Definition Step	7	.....	.....	.....
	03 = Destination Bank	3	.....	.....	.....
	04 = Generate Blob Image	Enabled	.....	.....	.....
	05 = Lower Threshold	25	.....	.....	.....
	06 = Upper Threshold	69	.....	.....	.....
	07 = Threshold Mode	Manual	.....	.....	.....
	08 = Min Blob Area (pixels)	5500	.....	.....	.....
	09 = Max Blob Area (pixels)	7500	.....	.....	.....
10	10 = Timeout (ms)	1000	.....	.....	.....
	11 = Discard Edge Blobs	Disabled	.....	.....	.....
	12 = Fill Holes in Blobs	Enabled	.....	.....	.....
	13 = Sort By	Horizontal Position	.....	.....	.....
	14 = Sorting Order	Ascending	.....	.....	.....
	15 = Sorting Point X	0	.....	.....	.....
	16 = Sorting Point Y	0	.....	.....	.....
	17 = Table Index	60	.....	.....	.....
	18 = Max Blobs in Table	3	.....	.....	.....
	Time of execution (µs)	0			
	01 = Number of Found Blobs	2			
	02 = Lower Threshold Used	25			
Setup	03 = Upper Threshold Used	75			
	Goto				
11	01 = Goto step	13	.....	.....	.....
	Time of execution (µs)	0			
	End				
12	01 = Start step	6	.....	.....	.....
	Time of execution (µs)	0			

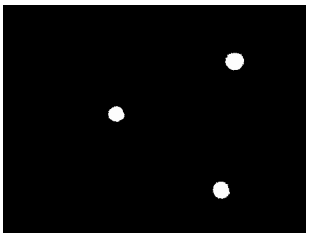
Rys. 9. Krok 10-12 – przetwarzanie obrazu cd.

Krok dziesiąty programu pokazany na rys. 9 umożliwia wyszukanie środków ciężkości trzech małych otworów naby na podstawie obrazu pokazanego na rys. 11.

a) b)



Rys. 10. a) obszar dużego otworu wykryty na zdjęciu obiektu, b) obszar naby wykryty na zdjęciu obiektu



Rys. 11. Obszar małych otworów wykrytych na zdjęciu obiektu

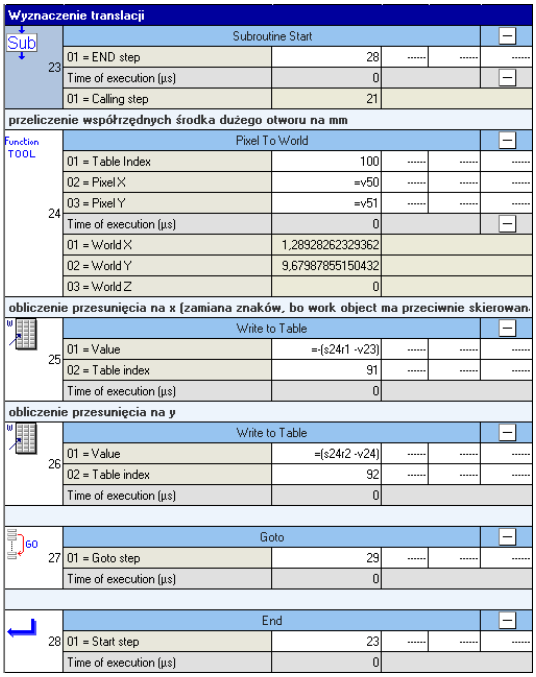
Położenia punktów środkowych wyznaczonych obszarów są zapisywane jako zmienne, które zostaną wykorzystane do obliczenia kąta obrotu i odległości przesunięcia detalu względem wzorca. Najpierw wyznaczana jest rotacja detalu (rys. 12).

Wyznaczenie rotacji					
	Subroutine Start				
13	01 = END step	22	.....	.....	.....
	Time of execution (µs)	1			
	01 = Calling step	11			
Obliczanie rotacji					
	For				
	01 = Start value	0	.....	.....	.....
	02 = End value	2	.....	.....	.....
14	03 = Increment	1	.....	.....	.....
	04 = END step	19	.....	.....	.....
	Time of execution (µs)	3			
	01 = Loop index	3			
Obliczanie kątów między środkiem dużego otworu, a środkami małych.					
	Distance and Angle				
	01 = Start point X	=v50	.....	.....	.....
	02 = Start point Y	=v51	.....	.....	.....
	03 = End point X	=v(60+s14r1*3)	.....	.....	.....
15	04 = End point Y	=v(61+s14r1*3)	.....	.....	.....
	Time of execution (µs)	36			
	01 = Distance	393,2374			
	02 = Line angle	53,46312			
zapis kąta					
	Write to Table				
16	01 = Value	=s15r2	.....	.....	.....
	02 = Table index	=70+s14r1*2	.....	.....	.....
	Time of execution (µs)	29			
zapis odległości między punktami					
	Write to Table				
17	01 = Value	=s15r1	.....	.....	.....
	02 = Table index	=71+s14r1*2	.....	.....	.....
	Time of execution (µs)	29			
Obliczanie różnicy kątów (między otrzymanym, a wzorcem)					
	Write to Table				
18	01 = Value	4r1*2)-v(40+s14r1*2)	.....	.....	.....
	02 = Table index	=80+s14r1	.....	.....	.....
	Time of execution (µs)	47			
	End				
19	01 = Start step	14	.....	.....	.....
	Time of execution (µs)	3			
obliczanie średniej rotacji					
	Write to Table				
20	01 = Value	=(v80+v81+v82)/3	.....	.....	.....
	02 = Table index	90	.....	.....	.....
	Time of execution (µs)	27			
	Goto				
21	01 = Goto step	23	.....	.....	.....
	Time of execution (µs)	1			
	End				
22	01 = Start step	13	.....	.....	.....
	Time of execution (µs)	0			

Rys. 12. Krok 18-22 – wyznaczenie rotacji.

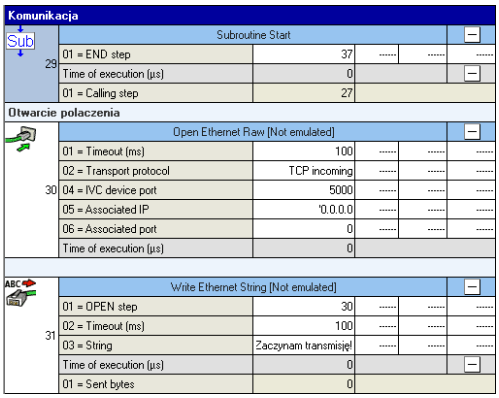
Z zastosowaniem opracowanego oprogramowania obliczany jest kąt zawarty pomiędzy prostą przechodzącą przez środek dużego otworu i małego otworu, a prostą równoległą do osi układu współrzędnych. Kąt jest obliczany dla wszystkich trzech małych otworów. Następnie obliczana jest różnica pomiędzy kątami zmierzonymi, a ich wartościami wzorcowymi. Średnia wartość tych różnic stanowi informacje o przesunięciu kątowym detalu

względem wzorca. Następnym krokiem jest wyznaczenie translacji (rys. 13).



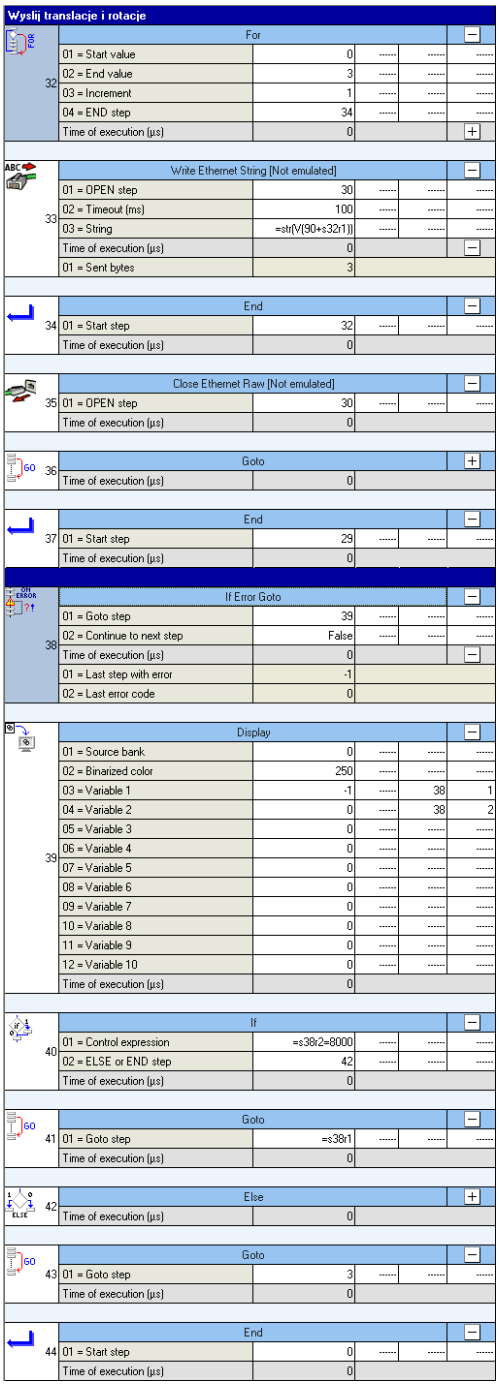
Rys. 13. Krok 23-28 – wyznaczenie wartości translacji

Określenie przesunięcia liniowego polega na obliczeniu różnicy położenia punktu środkowego dużego otworu analizowanego elementu i wzorca względem osi układu współrzędnych *xy*. Program przed wykonaniem obliczeń, umożliwia dokonanie konwersji zmiennych z pixeli na wartości w $[mm]$. Współczynniki konwersji są wyznaczane w procesie kalibracji, który został wykonany na etapie budowania stacji zrobotyzowanej. Po wykonaniu obliczeń dane są przesyłane do kontrolera robota (rys. 14 i 15).



Rys. 14. Krok 29-31 – komunikacja

Na początku programu *Komunikacja* otwierany jest port komunikacyjny połączenia Ethernet, zgodnie z zdefiniowanymi parametrami.



Rys. 15. Krok 32-44 – komunikacja cd.

Następnie dzięki zastosowaniu pętli *For* wyznaczone współrzędne są pojedynczo przesyłane do kontrolera robota. Po zakończeniu przesyłania transmisja jest zamykana i uruchamiany jest podprogram diagnostyczny *If Error Goto* (rys. 15), który sprawdza, czy całość programu przebiegła bez błędów. Jeśli w którymś kroku wystąpił błąd, czyli instrukcja programu nie została wykonana, program powróci do tego kroku, żeby wykonać go ponownie. Po poprawnym wykonaniu wszystkich instrukcji, program wraca na początek, a cykl jest powtarzany po wysłaniu sygnału wyzwalającego.

3.3 KONTROLER ROBOTA

Zadaniem kontrolera robota jest sterowanie całym procesem adaptacji układów odniesienia. Po zakończeniu pracy programu systemu wizyjnego kontroler odbiera dane. Odpowiada za to program zapisany w module *Komunikacja* przedstawiony poniżej.

Listing 2.

```
MODULE Komunikacja

PROC Komunikacja()

VAR socketdev polaczenie;
VAR socketdev klient_lan;
VAR string odczytany_ciag;
VAR num rotU;
VAR num transXU;
VAR num transYU;

SocketCreate polaczenie;

SocketConnect polaczenie, "192.168.100.102", 5000 \Time:=1;

odczytany_ciag:="";

SocketReceive polaczenie \Str:=odczytany_ciag;

TPWrite" rotacja workobjectu " + odczytany_ciag;

rotU:=odczytany_ciag;

SocketReceive polaczenie \Str:=odczytany_ciag;

TPWrite" translacja x " + odczytany_ciag;

tranXU:=odczytany_ciag;

SocketReceive polaczenie \Str:=odczytany_ciag;

TPWrite" translacja y " + odczytany_ciag;

tranYU:=odczytany_ciag;

SocketSend polaczenie \Str:="Odbior";
```

W programie zdefiniowano parametry połączenia oraz zmienne, do których zapisywane są otrzymane dane. Dodatkowo odebrane wartości translacji i rotacji układu są wyświetlane na ekranie panelu komunikacyjnego. Następnie są one wprowadzone do programu sterującego obróbką detalu.

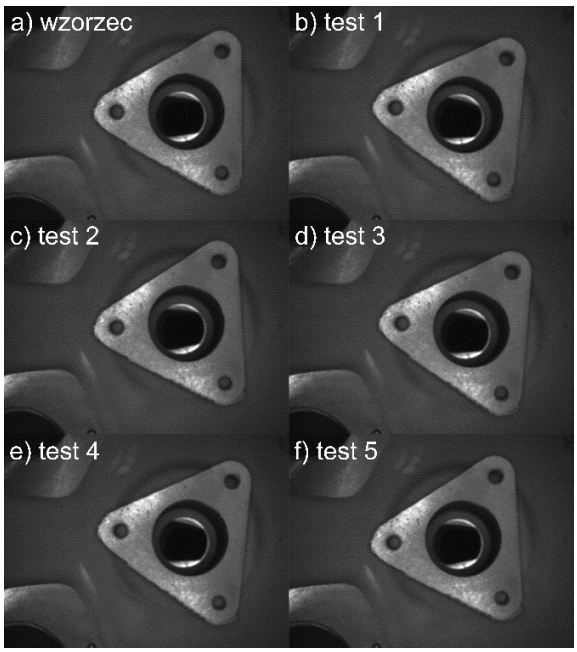
Listing 3.

```
PROC P_Naba_tr_cut()

W_Naba_tr_cut.oframe.trans.x := transXU;! przesunięcie układu na osi x
W_Naba_tr_cut.oframe.trans.y := transYU;! przesunięcie układu na osi y

object.rot := OrientZYX(rotU, 0, 0);

W_Naba_tr_cut.oframe.rot.z. := object.rot

MoveL T_Nab_tr_c_10,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

MoveL T_Nab_tr_c_20,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

MoveC
T_Nab_tr_c_30,T_Nab_tr_c_40,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

MoveL T_Nab_tr_c_50,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

MoveC
T_Nab_tr_c_60,T_Nab_tr_c_70,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

MoveL T_Nab_tr_c_80,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

 MoveC
T_Nab_tr_c_90,T_Nab_tr_c_10,v10,z1,Elektro_1\WObj:=W_Naba_tr_cut;

ENDPROC
```

Program przed wykonaniem instrukcji ruchu ramieniem robota aktualizuje pozycje i orientacje układu współrzędnych obiektu. Wykonywane następnie instrukcje ruchu stanowią ścieżkę, po której porusza się punkt TCP robota. Instrukcje odnoszą się do punktów w przestrzeni, zdefiniowane względem układu odniesienia obiektu. Adaptacja ustawienia tego układu współrzędnych umożliwia dopasowanie ścieżki punktu TCP do ustawienia obrabianego detalu.

4. BADANIA EKSPERYMENTALNE

W celu sprawdzenia poprawności działania przedstawionego rozwiązania, przeprowadzono testy dla pięciu elementów typu naba z dyfuzora silnika V2500.



Rys. 16. Obrazy pobrane w trakcie eksperymentu

Zmierzone wartości przemieszczenia i obrotu elementów wysłane do kontrolera robota, wyświetlono na panelu operatorskim. Wyniki przedstawiono w tabeli poniżej.

Tabela 1. Wyniki eksperymentu

Lp.	Nr testu	Translacja x [mm]	Translacja y [mm]	Rotacja [°]
1.	Test 1	-1,192649	-2,4328070	-5,995563
2.	Test 2	0,7901317	-0,0527715	3,0072373
3.	Test 3	-1,311357	-0,0020692	-4,994862
4.	Test 4	1,3764588	0,6453826	-5,679553
5.	Test 5	-3,067395	1,3629146	2,7604257

Na podstawie przeprowadzonych testów uzyskano korekcję translacji i orientacji układu odniesienia a przez to ścieżki robota z dokładnością 0,35 [mm] w każdej z osi. Otrzymana wartość wynika z dokładności robota. Dokładność korekcji trajektorii uzyskana z systemu wizyjnego jest o jeden rząd wielkości większa niż dokład-

ność robota. W prezentowanej pracy wykorzystano robota o powtarzalności 0.05 [mm] i dokładności 0,35 [mm]. Dokładność robota wynika z zastosowania opcji Absolute Accuracy, która jest potwierdzona raportem pomiarowym producentem pt. *Birth Certificate*.

5. WNIOSKI

W artykule przedstawiono autorskie rozwiązanie problemu adaptacji układu odniesienia przedmiotu dla zmiennej geometrii naby. Odchylenie geometrii określane jest przez program systemu wizyjnego SICK IVC-2DM1131, napisany w oprogramowaniu IVC Studio. Program na podstawie wyników analizy obrazu umożliwia obliczenie wartości translacji i rotacji naby, a nastepnie przesyła je do kontrolera IRC5 przez łącze Ethernet. Dla kontrolera robota opracowano autorskie oprogramowanie, w języku Rapid, umożliwiające odbieranie danych z systemu wizyjnego, które następnie są wykorzystane do modyfikacji układu odniesienia obiektu, w którym zdefiniowano ścieżkę obróbki. Przeprowadzono również testy weryfikacyjne zaproponowanego rozwiązania. Na podstawie prac badawczych stwierdzono ponadto, że ograniczenia dokładności realizacji trajektorii punktu TCP generowanej z wykorzystaniem systemu wizyjnego wynikają z dokładności robota.

Literatura

1. Burghardt A., Muszyńska M., Jagiełowicz-Ryznar C., Żylski W.: Aplikacja systemu wizyjnego do automatycznej adaptacji trajektorii narzędzia. W: XIV Konferencja Automatyzacji i Eksploatacji Systemów Sterowania i Łączności, ASMOR 2013, Jastrzębia Góra, 9-11 października 2013, , (materiały pokonferencyjne), s. 57-62.
2. Burghardt A., Szybicki D., Gierlak P., Kurc K., Muszyńska M.: Robotic automation of the turbo-propeller engine blade grinding process. DYNAMICAL SYSTEMS, Mechatronics and Life Sciences,2015, p. 121-130.
3. Burghardt A., Szybicki D., Kurc K., Muszyńska M.: Optimization of process parameters of edge robotic deburring with force control. "International Journal of Applied Mechanics and Engineering" 2016, vol.21, No.4, p.987-995, DOI: 10.1515/ijame-2016-0060.
4. Chen Y., Dong F.: Robot machining: recent development and future research issues. "The International Journal of Advanced Manufacturing Technology" 2013, p. 1-9.
5. Gierlak P., Burghardt A., Szybicki D., Szuster M., Muszyńska M.: On-line manipulator tool condition monitoring based on vibration analysis. "Mechanical Systems and Signal Processing" 2017, 89, p 14-26.
6. Horn K., Berthold P.: Robot vision. MIT Press, 1986.
7. Jain R., Kasturi R., Schunck B. G.: Machine vision. New York: McGraw-Hill, 1995.
8. Kuss A., Drust M., Verl A.: Detection of workpiece shape deviations for tool path adaptation in robotic deburring systems. "Procedia CIRP" 2016, 57, p. 545-550.
9. Muszyńska M., Burghardt A. ,Kurc K., Szybicki D.: Verification hybrid control of a wheeled mobile robot and manipulator. "Open Engineering" 2016, Vol. 6, Iss. 1, p. 64–72. DOI: 10.1515/eng-2016-0007.
10. Sioma A.: Programowanie systemów wizyjnych w środowisku IVC Studio [Programming of vision systems in the IVC Studio environment]. Katowice: Wyd. Kolumb, Katowice, 2010.