



Projekt aplikacji wspierającej konfigurowanie procesu paletyzacji kartonów dla robotów przemysłowych firmy Kawasaki — interfejs użytkownika

WOJCIECH KACZMAREK, BARTOSZ MOTYKA¹

Wojskowa Akademia Techniczna, Wydział Mechatroniki,
Instytut Systemów Mechatronicznych, 00-908 Warszawa, ul. S. Kaliskiego 2

¹ASTOR, 00-739 Warszawa, ul. Stępińska 22/30

Streszczenie. W artykule przedstawiono interfejs użytkownika proponowany do zautomatyzowanego procesu paletyzacji. Celem rozważań było zaproponowanie rozwiązania, które uprościłoby proces programowania robotów przemysłowych. Ze względu na zainteresowanie firmy ASTOR z Warszawy opracowano aplikację umożliwiającą zmianę konfiguracji procesu paletyzacji dla robotów firmy Kawasaki. Zaproponowane rozwiązanie, w połączeniu z opracowaną w ramach pracy dyplomowej w Instytucie Systemów Mechatronicznych Wojskowej Akademii Technicznej aplikacją do komunikowania się z robotem, umożliwia łatwe oraz intuicyjne programowanie robotów firmy Kawasaki bez konieczności znajomości ich języka programowania. Główną zaletą takiego rozwiązania jest udostępnienie operatorowi prostego w obsłudze środowiska, skracającego znacznie czas przekonfigurowania robota w przypadku zmiany parametrów paletyzacji (wielkość palety, rozmiary kartonów, liczba warstw itd.).

Słowa kluczowe: interfejs użytkownika, roboty przemysłowe, proces paletyzacji, programowanie robotów

Symbole UKD: 681.51

1. Wstęp

Robotyzacja jest wprowadzana do procesów przemysłowych od wielu lat. W latach siedemdziesiątych roboty wykorzystywano wyłącznie w produkcji wielkoseryjnej, ponieważ ich dostępność oraz samo przygotowanie zrobotyzowanych

stanowisk produkcyjnych było kosztowne, czasochłonne i ciągle jeszcze nie gwarantowało szybkiego zwrotu poniesionych kosztów. W dzisiejszych czasach, przy wykorzystaniu systemów wizyjnych, roboty zostały rozbudowane do maszyn potrafiących, pod wpływem zmian otoczenia, samodzielnie zmieniać algorytm swojego działania. Dynamiczny rozwój robotyzacji oraz wzrost dostępności robotów zwłaszcza w zastosowaniach przemysłowych sprawiły, że coraz częściej są one stosowane w produkcji krótkoseryjnej, lub nawet w produkcji jednostkowej [4].

Od kilku lat można zauważyć zmiany w liczbie robotów w poszczególnych gałęziach przemysłu. Do niedawna największy udział miał przemysł motoryzacyjny, elektrotechniczny i elektroniczny. W roku 2006 w przemyśle motoryzacyjnym zanotowano spadek sprzedaży o 17%, a przemyśle elektrotechnicznym i elektronicznym o 34%. Znacząco wzrosły natomiast inwestycje w branżę metalowej, chemicznej i spożywczej, gdzie w 2006 r. sprzedaż wzrosła o 25% [5, 7].

Duży wpływ na istniejącą w przemyśle sytuację ma kryzys finansowy. Zmusza on przedsiębiorców do obniżania kosztów produkcji bez jednoczesnych strat jakości produktów. Jednym ze sposobów może być przeniesienie produkcji do krajów nisko rozwiniętych, gdzie produkty mogą być wytwarzane taniej, ale nawet w tych regionach wymagania dotyczące kwalifikacji kadry i zarobków są z roku na rok coraz wyższe. Dlatego lepszym sposobem wydaje się być automatyzacja linii produkcyjnych. Po zakupie zrobotyzowanego stanowiska produkcyjnego, producent może pozwolić sobie na szybkie zmiany wolumenu produkcji (dziś produkując dużo, jutro mało bez ponoszenia dużych dodatkowych kosztów osobowych). Zwiększenie elastyczności z całą pewnością będzie zależało od dwóch czynników:

- elastyczności i prostoty oprogramowania, w jakie wyposażone są znajdujące się na zrobotyzowanej linii produkcyjnej maszyny,
- wykwalifikowanej siły roboczej (inżynierów-programistów).

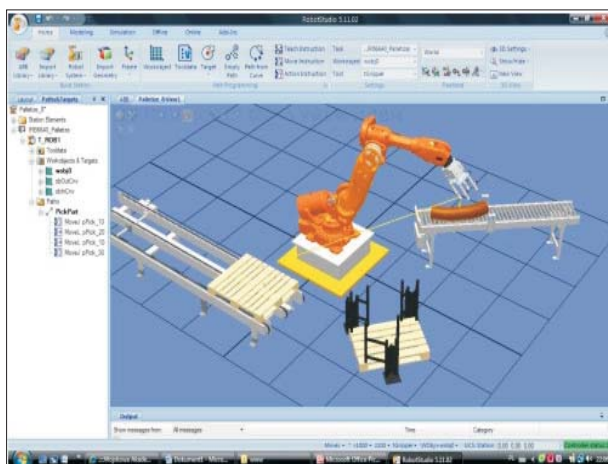
Oba te czynniki są ze sobą bardzo mocno związane. Jeśli środowiska do programowania robotów przemysłowych będą bardzo skomplikowane, konieczne będą zaawansowane, regularne i kosztowne szkolenia. Jeśli natomiast środowiska będą proste i intuicyjne, spełniając przy tym oczekiwania producenta, zmniejszy to znacznie koszty wyszkolenia kadry inżynierów [6].

2. Środowiska do programowania robotów przemysłowych

Gdy myślimy o producentach robotów przemysłowych, trudno doszukiwać się standaryzacji języków programowania. Każda z wiodących na rynku światowym firm rozwija swoje własne systemy, których programowanie jest realizowane w konkretnych językach programowania (np.: firma ABB — Rapid, Kawasaki — AS, Fanuc — Kail, Mitsubishi — MelfaBasic). Ponadto same interfejsy użytkowników oraz sposoby obsługi poszczególnych systemów znacznie się od siebie różnią. Efek-

tem takiego stanu rzeczy, jest zjawisko zakupu dla danej fabryki zrobotyzowanych stanowisk produkcyjnych jednego producenta, co nie zawsze jest najkorzystniejszym zjawiskiem, zwłaszcza ze względów ekonomicznych. Sensowne wydaje się więc poszukiwanie rozwiązań, które umożliwią nie tylko łatwą naukę programowania robotów przemysłowych, ale również uprościć proces przeprogramowania robotów na linii technologicznej w przypadku zmiany parametrów produkcji, oddając go ręce osób, które niekoniecznie muszą znać zaawansowane funkcje danego języka programowania.

Zarówno o pierwszym jak i o drugim rozwiązaniu myślą dzisiaj producenci robotów. Wprowadzone niedawno i prężnie rozwijane w ostatnim okresie rozbudowane środowiska do programowania robotów w trybie off-line (np.: ABB — RobotStudio — rys. 1, Kawasaki — PC-Roset, Fanuc — RoboGuide, Mitsubishi — Cosimir) pozwalają nie tylko na naukę programowania robotów danego producenta, ale również na znaczne skrócenie czasu programowania robotów. Są to jednak środowiska bardzo rozbudowane i poprawne ich wykorzystanie jest możliwe jedynie po odbyciu szeregu zazwyczaj kosztownych szkoleń.



Rys. 1. Ogólny widok środowiska RobotStudio v.5.10

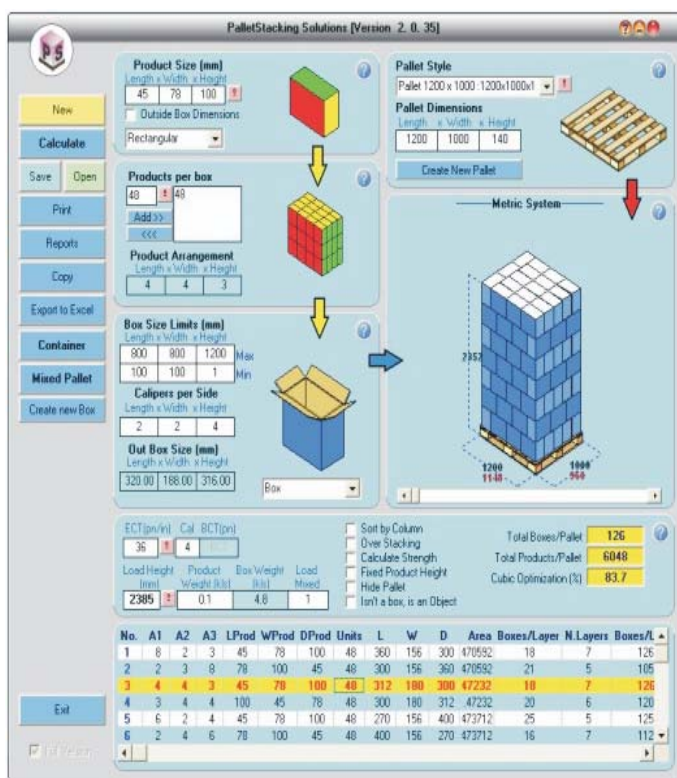
W przypadku drugiego rozwiązania można powiedzieć, iż istnieją dedykowane konkretnym procesom technologicznym (np.: spawanie łukowe, paletyzacja, cięcie laserem) środowiska, które pozwalają przede wszystkim na skrócenie czasu szkoleń operatorów oraz, dzięki swoim specyficznym zastosowaniom, uproszczenie samych środowisk programistycznych. Tworzenie takich aplikacji można podzielić na trzy etapy:

- opracowanie właściwego środowiska z odpowiednim interfejsem użytkownika (może być zrealizowane w dowolnym języku programowania),

- opracowanie nowego lub wykorzystanie istniejącego protokołu transmisji, umożliwiającego przesłanie niezbędnych do poprawnego funkcjonowania robota parametrów do jego kontrolera i przygotowanie aplikacji umożliwiającej przepływ danych w tym protokole,
- opracowanie elastycznej aplikacji w języku programowania danego robota i zaimplementowanie jej w jego kontrolerze.

3. Aplikacje wspierające konfigurowanie procesu paletyzacji

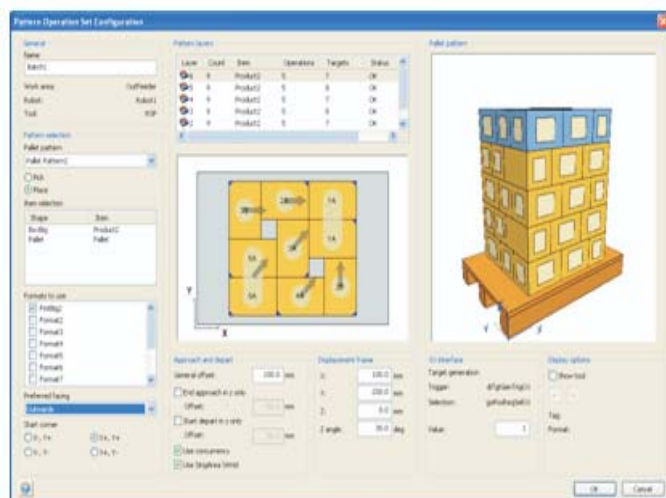
Do aplikacji wspierających proces paletyzacji można zaliczyć takie programy jak Pallet Stacking, QuickPalletMaker czy Cape Pack z modułem Pallet (rys. 2). Są one przeznaczone dla logistyków i technologów i nie można ich wykorzystać bezpośrednio do programowania robotów. Umożliwiają optymalne obliczenie wymiarów opakowań zbiorczych na podstawie wymiarów produktów, ustawienie opa-



Rys. 2. Ogólny wygląd programu Cape Pack

kowań na palecie i wypełnienie paletami kontenerów czy naczip. Zawierają w sobie bibliotekę znormalizowanych palet i kontenerów, co usprawnia pracę z programem. Wyniki obliczeń otrzymujemy w formie tekstowej i graficznej. Optymalizacja w ich przypadku wiąże się z umieszczeniem jak największej liczby produktów na jednej warstwie, a co za tym idzie i na całej palecie. Skutkuje to obniżeniem kosztów transportu produktów, co zwiększa dochody przedsiębiorstwa [15].

Innym typem środowisk umożliwiających programowanie robotów do procesu paletyzacji są aplikacje dedykowane dla operatorów obsługujących ten proces. Wiedzą niezbędną przy ich wykorzystaniu są jedynie wymiary palety i kartonów oraz liczba kartonów. Program pozwala na wybór schematu paletyzacji na podstawie wprowadzonych danych lub generuje go automatycznie (rys. 3).



Rys. 3. Ogólny widok programu PickMaster 5

Po zatwierdzeniu wybranych parametrów i nastaw program automatycznie generuje trajektorię ruchu robota i kąt podejścia dla każdego obiektu manipulacji. Aplikacja pozwala na paletyzację różnych rozmiarów produktów na jednej palecie i tych samych produktów na wielu paletach jednocześnie [22].

Przy wykorzystaniu tego typu aplikacji można skrócić czas przeprogramowania robotów do kilku minut, co w dzisiejszych czasach, przy produkcji krótkich serii danego asortymentu, jest niezmiernie ważne. Niestety na rynku jest niewiele tego typu środowisk. Ich ograniczenia, głównie ze względu na różne języki programowania robotów różnych producentów robotów, powodują, że to samo oprogramowanie można wykorzystać tylko dla jednej marki robotów, co znacznie ogranicza ich zastosowanie na liniach technologicznych fabryk.

Środowiska te są wykorzystywane zamiennie w zależności od zadania, jakie należy wykonać. Jeżeli celem będzie zaprogramowanie całego stanowiska do paletyzacji, na pewno najlepiej użyć symulatora robota, który pozwala na zaprogramowanie trajektorii robota i urządzeń peryferyjnych oraz zasymulować pracę całego stanowiska w środowisku do programowania robotów w trybie off-line (rys. 1). Natomiast gdy stanowisko już istnieje, a asortyment produktów często się zmienia, można wykorzystać aplikacje dedykowane bezpośrednio do procesu paletyzacji (rys. 3). Umożliwiają one zaprogramowanie robota do paletyzacji innego rodzaju produktów bez znajomości języka programowania robota. Aplikacje te same generują kod programu i wysyłają go do robota na podstawie wprowadzonych danych. Tego typu środowiska najczęściej wykorzystywane są w małych przedsiębiorstwach, gdzie produkty są paletyzowane w krótkich seriach i zachodzi potrzeba częstego przeprogramowania robota. Logistyk zajmujący się maksymalnym wypełnieniem palet wykorzysta programy optymalizacyjne, które znacznie ułatwią jego pracę.

4. Projekt aplikacji wspierającej konfigurowanie procesu paletyzacji kartonów dla robotów przemysłowych firmy Kawasaki — interfejs użytkownika

Z uwagi na brak oprogramowania dedykowanego bezpośrednio procesowi paletyzacji dla robotów Kawasaki oraz ze względu na zainteresowanie takim oprogramowaniem firmy ASTOR, będącej wyłącznym dystrybutorem tych robotów na Polskę, zdecydowano się podjąć próbę opracowania aplikacji, która ułatwiłaby uruchamianie stanowisk do paletyzacji przy wykorzystaniu robotów właśnie tej firmy. Aplikacja taka powinna umożliwić przeprogramowanie robota z dodatkowego panelu operatora (komputera) bez konieczności bezpośredniej ingerencji w jego program główny zaimplementowany w kontrolerze robota.

W ramach badań prowadzonych w Instytucie Systemów Mechatronicznych Wojskowej Akademii Technicznej, opracowano aplikację w środowisku Borland C++ Builder 6, której interfejs zapewnia wprowadzenie wszystkich danych niezbędnych do realizacji założonego procesu paletyzacji [3, 10]. Aplikacja składa się z czterech, występujących kolejno po sobie okien, które służą odpowiednio do:

- wprowadzenia danych dotyczących palety,
- wprowadzenia danych dotyczących kartonów,
- zdefiniowania sposobu ułożenia kartonów na palecie na warstwach: parzystej i nieparzystej,
- wprowadzenia danych niezbędnych do skomunikowania z robotem przemysłowym.

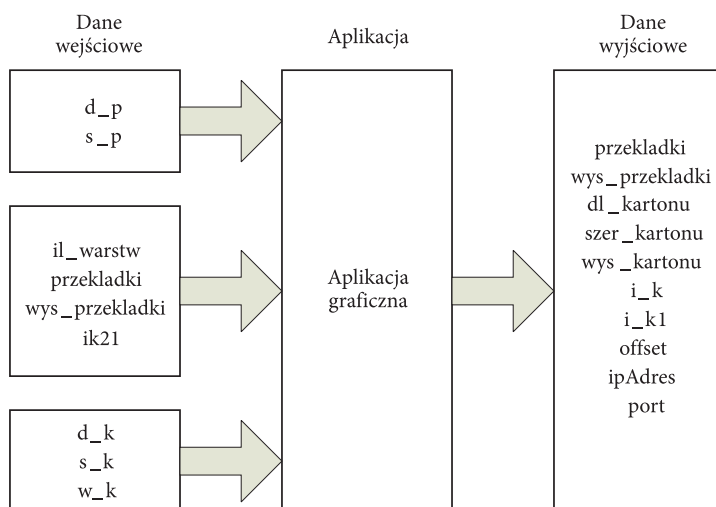
W dalszej kolejności program oblicza dane wyjściowe, formując je w odpowiednio zdefiniowaną strukturę, która zostaje udostępniona opracowanej aplikacji do komunikacji z robotem [1].

Zadaniem aplikacji jest ułatwienie programowania robotów do procesu paletyzacji, umożliwienie przeprogramowania robota do innego rodzaju asortymentu w jak najkrótszym czasie i bez znajomości języka programowania robota. W takim wypadku obsługa stanowiska ogranicza się tylko do wprowadzenia zmian w interfejsie i zatwierdzenia programu na TeachPendancie robota. Wyklucza to ingerencję pracownika (operatora stanowiska) w program robota, co ogranicza błędy spowodowane niewłaściwym zaprogramowaniem robota, a daje z kolei pewną elastyczność w przypadku zmiany produkowanego asortymentu. Aplikacja umożliwi również uruchomienie programu z wprowadzonymi zmianami z ograniczoną prędkością w celu sprawdzenia poprawności ruchów robota.

4.1. Schemat funkcjonalny programu

Zadaniem programu jest udostępnienie zmiennych niezbędnych do procesu paletyzacji, aplikacji służącej do komunikacji z robotem. W tym celu program pobiera dane wejściowe, które wprowadza użytkownik, przetwarza je i w odpowiedniej formie przekazuje aplikacji przesyłającej te dane do kontrolera robota. Dane wejściowe zostały podzielone na trzy części (rys. 4):

- dane dotyczące palety — wymiary palety; liczba warstw kartonów na palecie; zmienna określająca liczbę warstw, po których występuje przekładka; wysokość przekładki i występowanie warstw nieparzystych i parzystych,



Rys. 4. Schemat funkcjonalny aplikacji

- dane dotyczące kartonów — wymiary kartonów; liczba kartonów na warstwie parzystej i nieparzystej oraz kąt podejścia,
- dane dotyczące parametrów komunikacyjnych robota — numer IP robota; numer portu robota i prędkość robota.

Opis oraz typ zmiennej danych wejściowych przedstawiono w tabeli 1, natomiast opis oraz typ zmiennych danych wyjściowych zawiera tabela 2.

TABELA 1

Typ zmiennych danych wejściowych

Opis zmiennej	Oznaczenie zmiennej	Typ zmiennej
długość palety	d_p	AnsiString
szerokość palety	s_p	AnsiString
liczba warstw kartonów na palecie	il_warstw	AnsiString
zmienna oznaczająca liczbę warstw, po których występuje przekładka	przekladki	integer
wysokość przekładki	wys_przekladki	integer
występowanie warstw nieparzystych i parzystych	ik21	integer
długość kartonów	d_k	AnsiString
szerokość kartonów	s_k	AnsiString
wysokość kartonów	w_k	AnsiString
liczba kartonów na warstwie nieparzystej	i_k	AnsiString
liczba kartonów na warstwie parzystej	i_k1	AnsiString
liczba określająca kąt podejścia	Offset	integer
pierwsza część numeru IP robota składająca się z trzech znaków	iip0	AnsiString
druga część numeru IP robota składająca się z trzech znaków	iip1	AnsiString
trzecia część numeru IP robota składająca się z trzech znaków	iip2	AnsiString
czwarta część numeru IP robota składająca się z trzech znaków	iip3	AnsiString
numer portu robota	Port	AnsiString
prędkość robota w procentach	cpredkosc	AnsiString

TABELA 2

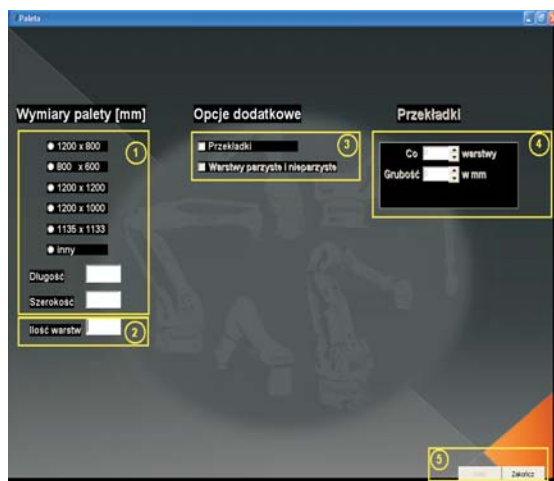
Typ zmiennych danych wyjściowych

Opis zmiennej	Oznaczenie zmiennej	Format zmiennej
Adres IP robota	przekladki	integer
Numer portu	wys_przekladki	integer
Prędkość robota	dlugosc_kartonu	integer
Ilość kartonów w warstwie nieparzystej	szerokosc_kartonu	integer
Ilość kartonów w warstwie parzystej	wysokosc_kartonu	integer
Ilość warstw	i_k	integer
Wysokość kartonu	i_k1	integer
Numer offsetu	offset	integer
Zmienna określająca ilość warstw, po których występuje przekładka	ipAdres	AnsiString
Wysokość przekładki	port	AnsiString
Długość kartonu	cpredkosc	AnsiString
Szerokość kartonu	il_w	integer
Współrzędne X kartonów warstwy nieparzystej	iksy[0...ik]	integer
Współrzędne Y kartonów warstwy nieparzystej	igrek[0...ik]	integer
Obroty wokół osi Z kartonów warstwy nieparzystej	obroty[0...ik]	integer
Współrzędne X kartonów warstwy parzystej	iksy1[0...ik1]	integer
Współrzędne Y kartonów warstwy parzystej	igrek1[0...ik1]	integer
Obroty wokół osi Z kartonów warstwy parzystej	obroty1[0...ik1]	integer

4.2. Interfejs aplikacji

Interfejs graficzny składa się z czterech okien. W każdym z nich użytkownik wprowadza dane i wybiera ustawienia niezbędne do przeprowadzenia procesu paletyzacji. Pierwsze okno pozwala wybrać jeden z wymiarów palety lub podać własne, gdy paletyzacja będzie odbywała się na niestandardowej paletce (rys. 5, pole 1).

Następnie należy podać liczbę warstw kartonów na paletce (rys. 5, pole 2). Wśród opcji dodatkowych użytkownik ma do wyboru „Przekładki” i „Warstwy parzyste i nieparzyste” (rys. 5, pole 3). Zaznaczenie opcji „Przekładki” uaktywnia panel „Przekładki”, gdzie wprowadzana jest liczba warstw, po których ma wystąpić przekładka oraz grubość przekładki w milimetrach. Domyślnie zmienne te przyjmują

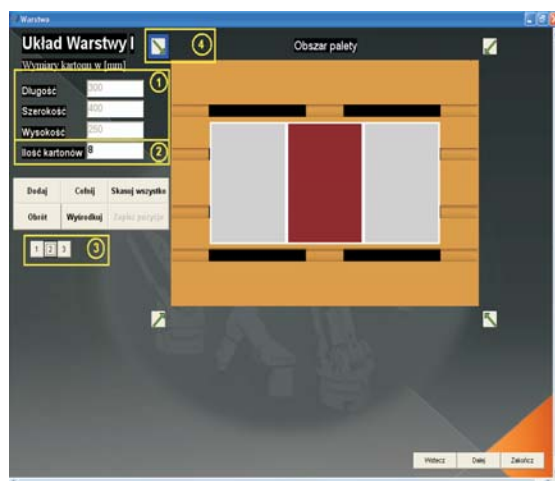


Rys. 5. Widok pierwszego okna aplikacji

wartość 1, co oznacza, że przekładka będzie układana pomiędzy każdą warstwą, a jej grubość wynosi jeden milimetr (rys. 5, pole 4). Zaznaczenie opcji „Warstwy parzyste i nieparzyste” umożliwia wprowadzenie innej konfiguracji warstwy parzystej i innej konfiguracji warstwy nieparzystej. Możliwe jest to w oknie drugim i trzecim aplikacji. Jeżeli opcja ta nie zostanie zaznaczona okno trzecie będzie pominięte podczas całego procesu konfiguracji. Oznacza to, że kartony na wszystkich warstwach będą układane według tego samego schematu. Po wprowadzeniu niezbędnych danych użytkownik może przejść do kolejnego okna naciskając przycisk „Dalej” (rys. 5, pole 5).

Drugie okno programu przedstawia paletę w wcześniej wybranych wymiarach. Użytkownik może wpisać w polach edycyjnych wymiary kartonu (rys. 6, pole 1), a następnie liczbę kartonów w warstwie (rys. 6, pole 2). Następnie naciskając przycisk „Dodaj”, w górnym lewym rogu palety pojawia się pierwszy karton o podanych wymiarach, przeskalowany względem rozmiaru palety. Automatycznie zostaje zablokowana możliwość zmiany rozmiaru kartonu oraz pojawia się przycisk z numerem kartonu (rys. 6, pole 3). Przesunięcie kartonu jest możliwe po najechaniu kursorem na obszar kartonu i naciśnięciu lewego przycisku myszki. Użytkownik może dodać wszystkie kartony bez ich rozmieszczania, a następnie, naciskając przycisk „Wyśrodkuj”, automatycznie wyśrodkować wszystkie kartony. Jeżeli liczba dodanych kartonów jest za duża i nie mieszczą się one na paletce, użytkownik może usunąć pojedynczo kartony, naciskając przycisk „Cofnij”. Funkcja wyśrodkowania automatycznie kasuje kartony, które nie mieszczą się na paletce, i wyświetla informujący o tym komunikat (rys. 7).

Aby zmienić rozmiar kartonu, należy usunąć wszystkie kartony z palety. Można to zrobić pojedynczo, naciskając przycisk „Cofnij”, lub skasować wszystkie kartony, naciskając przycisk „Skasuj wszystko”. Po usunięciu ostatniego kartonu pola edycyjne wymiarów kartonów zostają odblokowane. Aby obrócić karton o 90°, należy



Rys. 6. Drugie okno aplikacji

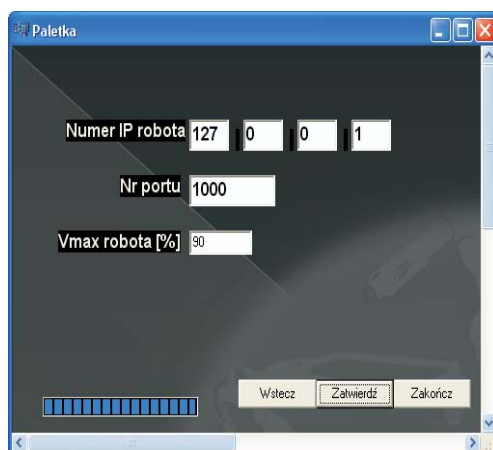


Rys. 7. Okno aplikacji z komunikatem

go zaznaczyć, naciskając przycisk z jego numerem, a następnie nacisnąć przycisk „Obrót”. Po ustawieniu kartonów na paletę użytkownik może wybrać kąt podejścia robota podczas paletyzowania kartonów. Możliwe jest to dzięki przyciskom rozmieszczonym wokół palety (rys. 6, pole 4). Po naciśnięciu jednego z czterech przycisków zostanie on podświetlony na niebieski kolor.

Po wpisaniu i zaznaczeniu wszystkich parametrów udostępniony (aktywowany) zostaje przycisk „Zapisz pozycje”. Naciskając go, użytkownik zapisuje współrzędne kartonów względem palety, odblokowując jednocześnie przycisk „Dalej”, który umożliwia przejście do kolejnego okna programu.

Jeżeli użytkownik zaznaczył w pierwszym oknie opcję „warstwy parzyste i nieparzyste” kolejnym oknem będzie okno trzecie. Umożliwia ono zdefiniowanie i zapisanie schematu warstwy parzystej, w analogiczny sposób jak miało to miejsce dla warstwy nieparzystej (drugie okno aplikacji). Natomiast, jeżeli użytkownik nie zaznaczył opcji różnych warstw w pierwszym oknie, kolejnym oknem będzie okno parametrów robota (rys. 8). W tym oknie użytkownik podaje IP robota oraz numer portu, na który zostaną przesłane dane niezbędne do procesu. Następnie wpisuje prędkość robota w zakresie od 0 do 100 procent w zależności od przenoszonych rodzajów produktów i ich bezwładności. Mała prędkość robota może być wykorzystana podczas testowania trajektorii robota po wprowadzeniu nowych zmiennych.



Rys. 8. Okno parametrów komunikacyjnych robota

Dane do robota można wysłać, naciskając przycisk „Zatwierdź”, a ich zaawansowanie przedstawia pasek postępu znajdujący się u dołu okna. W każdym momencie aplikacja może zostać zakończona przez użytkownika po naciśnięciu przycisku „Zakończ”, natomiast za pomocą przycisku „Wstecz” można cofnąć się do poprzedniego okna, aby zmienić wybrane tam parametry.

Opracowana aplikacja została przetestowana dwuetapowo. W pierwszym etapie wykonano to w trybie symulacji komputerowej, wykorzystując do tego celu aplikacje klienta i serwera oraz korzystając z możliwości środowiska do programowania robotów firmy Kawasaki w trybie off-line PC-Roset. W drugim etapie, już poprawnie działającą aplikację przetestowano razem z równoległe opracowywaną aplikacją do przesyłania przygotowanych w protokole UDP danych do robota Kawasaki oraz właściwą aplikacją do paletyzacji zaimplementowaną w kontrolerze robota [8, 9, 11]. Testy praktyczne przeprowadzono w firmie AB Industry w Ożarowie Mazowieckim (rys. 9).



Rys. 9. Testy proponowanej aplikacji w firmie AB Industry

5. Podsumowanie

Realizację przedstawionych w artykule rozważań poprzedziła rozmowa z pracownikami firmy ASTOR Sp. z o.o, która jest wyłącznym dystrybutorem robotów Kawasaki na Polskę. W czasie tej rozmowy okazało się, iż istnieje zapotrzebowanie na aplikacje dedykowane konkretnym procesom technologicznym dla robotów Kawasaki. Oprogramowanie takie mogłoby być dołączane do sprzedawanego stanowiska bez dodatkowych kosztów. Inną alternatywą mogłoby być dołączanie środowiska Pc-Roset do programowania robotów w trybie off-line, jednak to rozwiązanie byłoby obciążone dodatkowymi kosztami.

Opracowane oprogramowanie, dzięki życzliwości firmy AB Industry z Ożarowa Mazowieckiego, zostało przetestowane przy wykorzystaniu robota Kawasaki (rys. 9). Zaproponowane rozwiązanie w połączeniu z aplikacją do komunikacji umożliwia zaprogramowanie robota do paletyzacji kartonów w krótkim czasie, w prosty sposób. Przy wykorzystaniu aplikacji do komunikacji z robotami innych producentów (np.: Fanuc, ABB) aplikacja może zostać wykorzystana także do programowania robotów innych marek. Biorąc pod uwagę fakt, że na rynku dostępnych jest niewiele aplikacji tego typu, a roboty przemysłowe są coraz częściej wykorzystywane do procesu paletyzacji w różnych gałęziach przemysłu, aplikacja może znaleźć wielu nabywców, zwłaszcza tam gdzie paletyzacja krótkich serii asortymentu zmusza do częstych zmian w oprogramowaniu robota.

Zaprezentowana aplikacja może być rozbudowana o moduł programowania robota do procesu depaletyzacji, co rozszerzy jej możliwości zastosowania. Można również umieścić w programie zestaw gotowych do wykorzystania schematów ułożenia kartonów na poszczególnych warstwach. Następnym krokiem rozbudowy może być wizualizacja stanu faktycznego procesu paletyzacji na ekranie monitora. Wraz z jego postępowaniem pojawiałyby się na ekranie kolejne kartony uło-

żone na palecie. Ponadto wizualizacja mogłaby zostać przedstawiona na ekranie w przestrzeni trójwymiarowej. Aby ułatwić pracę z aplikacją na linii produkcyjnej, celowe wydaje się umieszczenie panelu dotykowego. Dzięki temu ze stanowiska zostanie wyeliminowana myszka i klawiatura, które w takich warunkach często ulegają zniszczeniu.

Artykuł wpłynął do redakcji 6.08.2009 r. Zweryfikowaną wersję po recenzji otrzymano w listopadzie 2009 r.

LITERATURA

- [1] *AS Language Reference Manual*, Kawasaki Heavy Industries LTD, 2007.
- [2] P. DUTKIEWICZ, W. WRÓBLEWSKI, *Modelowanie i sterowanie robotów*, PWN, Warszawa, 2003.
- [3] M. FLENOV, *C++ Elementarz hakera*, Helion, 2005.
- [4] K. GROHS, *Robotyzacja lekarstwem na kryzys?*, www.automatykab2b.pl
- [5] J. HONCZARENKO, *Roboty przemysłowe. Elementy i zastosowanie*, WNT, Warszawa, 1992.
- [6] W. KACZMAREK, *Elementy robotyki przemysłowej*, WAT, Warszawa, 2008.
- [7] J. MOŻARYN, Z. PIĄTEK, *Rynek robotyki przemysłowej bez tajemnic*, www.automatykab2b.pl
- [8] *PC-Roset Ligot — Pierwsze kroki*, ASTOR Sp. z. o.o.
- [9] *Programowanie w języku AS*, ASTOR Sp. z. o.o.
- [10] A. STASIEWICZ, *C++ Builder. Całkiem inny świat*, Helion, 2005.
- [11] *TCP/IP Communication Manual*, Kawasaki Heavy Industries LTD, 2007.
- [12] K. TOMASZEWSKI, *Roboty przemysłowe*, WNT, Warszawa, 1993.

W. KACZMAREK, B. MOTYKA

Project of application supporting configuration of palletization process of packets for Kawasaki industrial robots — user's interface

Abstract. In this paper, the problems of programming industrial robots were presented. The authors proposed user's interface to show the destination of application development. Creating new applications give the possibility of the programming process simplification and programming time reduction.

Keywords: user's interface, industrial robots, palletization process, robots programming

Universal Decimal Classification: 681.51