

Odczyt kodów felg samochodowych w procesie produkcyjnym

Jacek Dunaj

Przemysłowy Instytut Automatyki i Pomiarów PIAP

Streszczenie: W artykule przedstawiono sposób realizacji odczytu kodów felg samochodowych. Opracowane stanowisko zostało wdrożone przez pracowników Przemysłowego Instytutu Automatyki i Pomiarów PIAP w zakładach firmy ATS Stahlschmidt & Maiworm w Stalowej Woli. W artykule omówiono zastosowane algorytmy identyfikacji kodów oraz przedstawiono problemy, jakie napotkano podczas realizacji tej pracy.

Słowa kluczowe: FelgenCode, laserowy czujnik pomiaru odległości, metoda najmniejszych kwadratów

DOI: 10.14313/PAR_204/83

1. Wprowadzenie

W 2008 r. Przemysłowy Instytut Automatyki i Pomiarów PIAP uruchomił w fabryce firmy ATS Stahlschmidt & Maiworm w Stalowej Woli (należącej obecnie do grupy Uniwheels) stanowisko do odczytu kodów felg samochodowych. Od tego czasu oprogramowanie stanowiska było kilkakrotnie modernizowane, co uczyniło je jedną z ciekawszych aplikacji przemysłowych wykonanych przez Instytut.



Rys. 1. Odlew felgi samochodowej z kodem FelgenCode V1.3 firmy Numtec

Fig. 1. Car's wheels casting with Numtec's FelgenCode ver. V1.3

Zakłady ATS w Stalowej Woli produkują aluminiowe felgi do samochodów znanych europejskich marek. Ponieważ w fabryce jest kilkadziesiąt stanowisk odlewniczych,

a gotowe odlewy trafiają na wspólne transportery, dużym utrudnieniem była ich identyfikacja przed dalszą obróbką. Dla rozwiązania tego problemu każde koło jest odlewane wraz z dodatkowym kodem cyfrowym FelgenCode firmy Numtec Interstahl opracowanym specjalnie dla znakowania felg (rys. 1).

Kod ten, usuwany na końcu procesu technologicznego, ma stałe wymiary i określony format. W chwili przekazywania stanowiska do eksploatacji stosowano 12-bitowy kod w wersji V1.3 (rys. 2a), umożliwiający znakowanie odlewów numerami od 0 do 4095. W 2009 r. rozpoczęto stosowanie 14-bitowego kodu w wersji V3.0 (rys. 2b), który pozwala na oznaczanie numerami od 5000 do 21 383. Od wprowadzenia kodu V3.0 obydwa sposoby znakowania są w bieżącej produkcji stosowane równolegle.

Kody w wersjach V1.3 i V3.0 mają wiele cech wspólnych. W obu przypadkach dwa skrajne karby od lewej strony: szeroki i wąski są karbami referencyjnymi wyznaczającymi początek pola kodowego. W wersji V1.3 kodu wewnątrz pola znajduje się kolejny szeroki karb referencyjny, w wersji V3.0 kodu występują dwa takie karby. W obu przypadkach pole kodowe jest zakończone pojedynczym szerokim karbem referencyjnym. Karby referencyjne są tymi elementami, które zawsze występują w polu kodowym, niezależnie od numeru przypisanego danej feldzie. Wąski karb lub brak wąskiego karbu w miejscach oznaczonych na rys. 2 cyframi od 0 do 13 oraz S1, S2 i S3 określają wartości „1” lub „0” danego bitu kodu lub odpowiedniej sumy kontrolnej. Wszystkie karby i odstęp między karbami (lub odstęp między miejscami, w których karby mogą wystąpić) mają ściśle określoną szerokość i wysokość mierzoną względem powierzchni bocznej odlewu.

2. Układ sterująco-pomiarowy urządzenia

Początkowo rozważana była budowa stanowiska na bazie systemu wizyjnego wykorzystującego tzw. kamerę liniową oraz specjalizowany program NeuroCheck stoso-

Zobacz więcej

Pobierz bezpłatną aplikację PAR+
App Store | Google Play

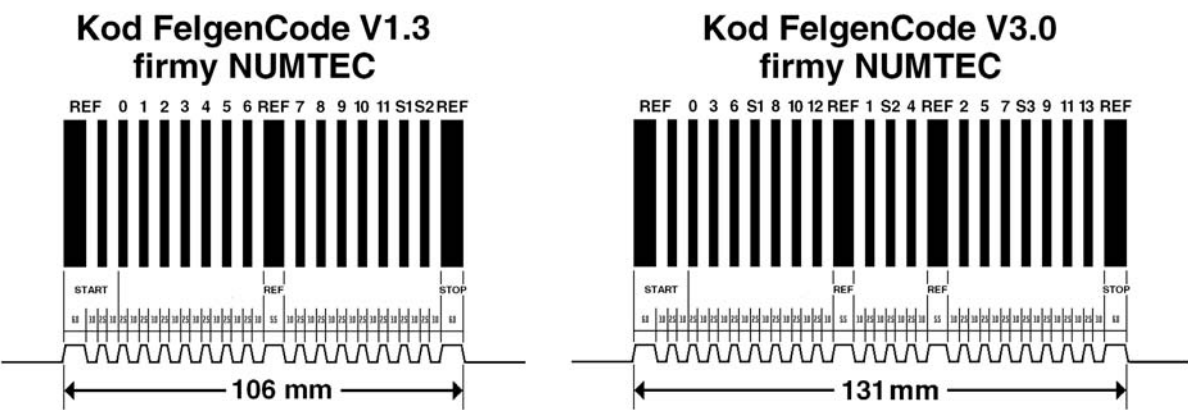


wany we wcześniejszych aplikacjach. Po wykonaniu szeregu testów koncepcję tę zarzucono ze względu na szacowany wysoki koszt takiego rozwiązania oraz obawy, że urządzenie wizyjne może być zbyt delikatne, jak na agresywne warunki pracy w pobliżu odlewni. Zdecydowano się na zastosowanie laserowego czujnika pomiaru odległości. Idea jego stosowania polega na zamocowaniu sondy pomiarowej na obrotowym ramieniu (rys. 4) i przemieszczaniu jej ze stałą prędkością kątową i w stałej odległości od powierzchni bocznej felgi. Urządzenie najpierw obraca ramieniem, aby wykryć pole kodowe, a następnie usta-

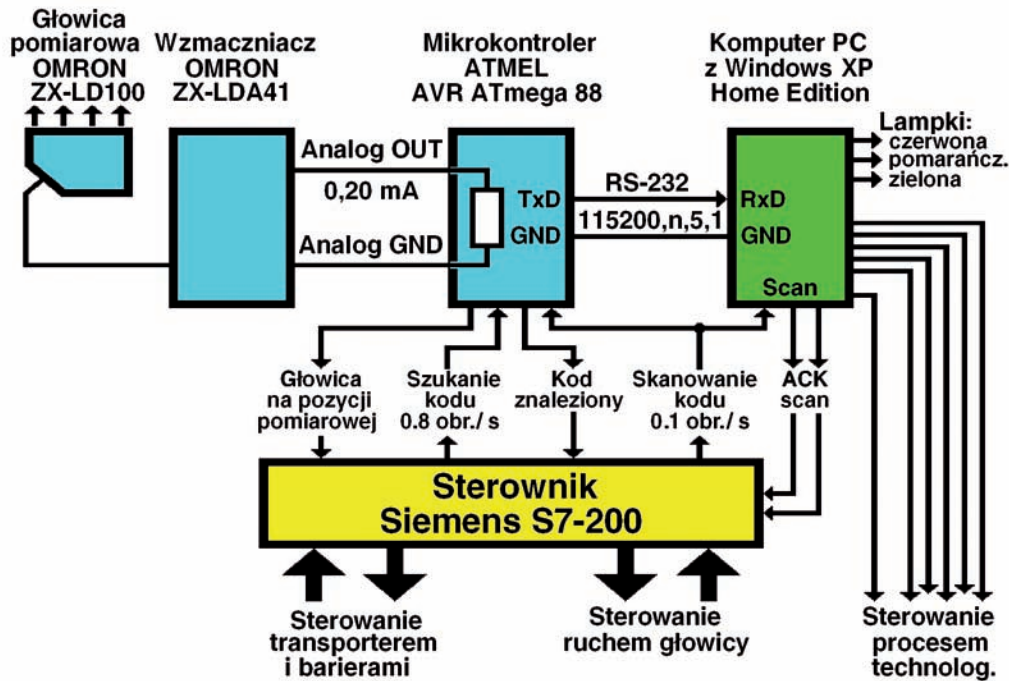
wia sondę za jego końcem i powtórnie przemieszcza ją od ostatniego do pierwszego karbu referencyjnego. Podczas tego ruchu wartości kolejnych pomiarów są na bieżąco przesyłane interfejsem RS-232 do komputera PC, gdzie są zapamiętywane, a po zakończeniu skanowania – podane analizie.

Układ sterująco-pomiarowy urządzenia (rys. 3) składa się z następujących elementów:

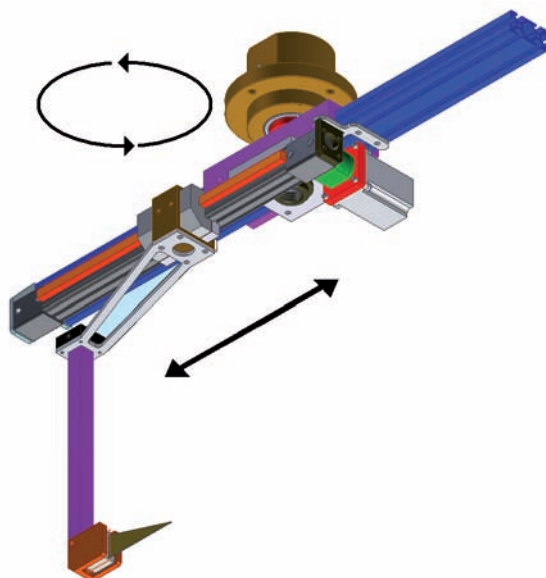
- laserowej głowicy pomiarowej ZX-LD100 firmy Omron o zasięgu od 100 do 140 milimetrów podłączonej do wzmacniacza ZX-LDA41. Wzmacniacz posiada prądowe



Rys. 2. Kody FelgenCode firmy Numtec w wersjach V1.3 i V3.0 do znakowania felg samochodowych w procesie produkcji
Fig. 2. Format of Numtec's FelgenCode ver 1.2 and 3.0 for marking car's wheels during manufacturing process



Rys. 3. Układ sterująco-pomiarowy urządzenia
Fig. 3. Scheme of control and measurement system



Rys. 4. Zespół ramienia z głowicą pomiarową
Fig. 4. Arm unit with laser distance sensor

wyjście analogowe, sprzężone z wejściem analogowym mikrokontrolera AVR ATmega88.

- sterownika przemysłowego Siemens S7-200 sterującego dwoma silnikami krokowymi do poruszania ramieniem, na którym zamocowano głowicę pomiarową. Jeden z tych silników przemieszcza głowicę wzdłuż promienia felgi, drugi powoduje jej ruch nad powierzchnią felgi (rys. 4).

Dodatkowo S7-200 steruje ruchem transportera do przemieszczania felg oraz barierami zabezpieczającymi.

- komputera przemysłowego PC firmy Advantech pracującego pod kontrolą systemu operacyjnego Windows XP Home Edition i wyposażonym w standardowy interfejs szeregowy RS-232 oraz dodatkową kartę WE-WY dwustanowych firmy Wasco. Osadzona na tym komputerze aplikacja identyfikuje kody felg oraz obsługuje bazy danych i arkusze kalkulacyjne związane z wykonywanym zadaniem.

3. Zasada działania urządzenia

Sterownik Siemens S7-200 sterując transporterem przemieszcza odlew do wnętrza stanowiska i przy pomocy pneumatycznego zespołu pozycjonującego ustawia felgę współosiowo z zespołem ramienia głowicy pomiarowej (rys. 5). Następnie głowica jest przemieszczana w kierunku powierzchni felgi w taki sposób, aby znalazła się wewnątrz swojego zakresu pomiarowego. Z zestawem pomiarowym firmy Omron bezpośrednio współpracuje mikrokontroler AVR ATmega88, który sygnalizuje sterownikowi S7-200 osiągnięcie przez czujnik laserowy pozycji roboczej. W tym momencie układ sterowania uruchamia ruch ramienia z głowicą nad powierzchnią felgi z prędkością równą 0,8 obrotu na sekundę. W trak-



Rys. 5. Stanowisko odczytu kodów felg w działaniu
Fig. 5. Car's wheels codes scanning station

cie tego ruchu program mikrokontrolera AVR ATmega88 identyfikuje położenie pola kodowego felgi i przekazuje tę informację do sterownika S7-200.

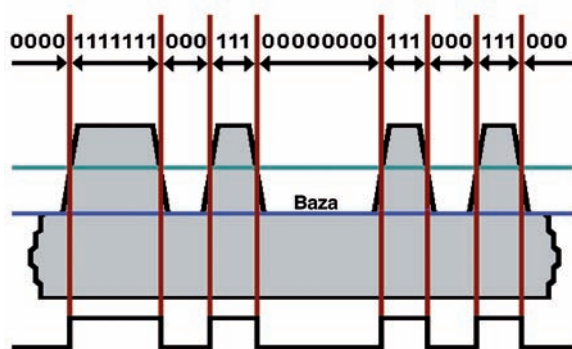
Sterownik ustawia ramię z czujnikiem w pozycji za ostatnim karbem referencyjnym kodu i rozpoczyna jego obrót nad kodem felgi z prędkością równą 0,1 obrotu na sekundę. Ruch ten jest sygnalizowany dwustanowym sygnałem sterującym do mikrokontrolera i komputera przemysłowego PC. W trakcie jego wykonywania program mikrokontrolera AVR ATmega88 odczytuje informację z wyjścia analogowego wzmacniacza ZX-LDA41, koduje ją, a następnie transmituje do komputera PC, gdzie jest ona zapamiętywana do dalszej analizy.

Samą analizę kodu felgi aplikacja komputera PC wykonuje bezpośrednio po zakończeniu skanowania, a następnie za pomocą dwóch sygnałów WY do sterownika S7-200 wydaje polecenie powtórnego odczytu (w razie jego niepowodzenia), bądź przemieszczenia felgi poza stanowisko skanujące. Kolejny odczyt może być poprzedzony mechaniczną próbą poprawienia pozycjonowania.

Należy nadmienić, że w jednym z wcześniejszych wariantów stanowiska odczyt i transmisję danych ze wzmacniacza realizował dodatkowy moduł ZX-SF11 firmy Omron. Niestety, protokół transmisyjny zaimplementowany w tym urządzeniu działał w trybie pytanie – odpowiedź, a sam komunikat był bardzo rozbudowany, czego skutkiem była niska przepustowość kanału transmisyjnego sięgająca zaledwie ok. 120 pomiarów na cały skanowany kod. Przejście zadań modułu ZX-SF11 przez mikrokontroler AVR ATmega88 i opracowanie własnego, prostszego protokołu transmisyjnego zwiększyło liczbę przekazywanych pomiarów do ponad 5500 na kod.

4. Idea odczytu kodu felgi realizowana przez aplikację komputera PC

Odczyt kodu felgi polega na wyznaczeniu bazowej odległości sondy od powierzchni felgi i odniesieniu wszystkich zapamiętanych pomiarów do wyznaczonej bazy. Następnie wyniki pomiarów zostają przekształcone na ciąg zer i jedynek w taki sposób, że jeśli różnica wartości bazowej i wartości danego pomiaru jest większa niż określona część wysokości karbu kodu, to danemu pomiarowi zostaje przypisana wartość „1”, w przeciwnym razie wartość „0”. Grupy kolejnych „1” i „0” wyznaczają obecność kolejnych karbów kodu i szczelin między karbami (rys. 6).



Rys. 6. Idea analizy kodu felgi
Fig. 6. How car's wheels codes are analyzed

Podstawowym zadaniem stanowiska jest identyfikacja kodów felg o średnicach od 15 do 20 cali. Przy stałej prędkości kątowej ramienia z głowicą pomiarową większa średnica odlewu oznacza większą prędkość liniową głowicy względem pola kodowego, a więc mniejszą gęstość pomiarów na jednostkę długości. Samo ramię nie obraca się z jednakową prędkością w całym zakresie pomiarowym, ponieważ na początku i na końcu tego ruchu występują zmiany prędkości związane z rozpędzaniem i hamowaniem. Ponadto stanowisko odczytu pracuje na początku linii technologicznej, gdzie jeszcze nie obrobione odlewy nie są idealnie symetryczne, więc można je tylko zgrubnie pozycjonować względem osi obrotu ramienia.

Należy nadmienić, że oprogramowanie stanowiska odczytu nie dysponuje żadną informacją czy aktualnie skanowana felga ma kod w wersji V1.3 czy w wersji V3.0. Oznacza to, że aplikacja komputera PC równolegle musi realizować algorytmy identyfikujące obydwa rodzaje kodów. Kryteria identyfikacji są jednak na tyle dokładne, że eliminują przypadek zasygnalizowania przez program prawidłowego odczytu kodu w wersji, która nie dotyczy danej felgi.

Z tymi problemami musiała poradzić sobie aplikacja komputera PC do identyfikacji kodów felg. Dlatego rozbudowano ją o kilka funkcji umożliwiających testowanie skuteczności odczytu przy różnych wariantach usta-

wień. Ich realizacja umożliwia m.in. rejestrację do plików wyników pomiarów wykonanych podczas skanowania felg a następnie wykorzystania ich do testowania aplikacji w trybie off-line. Aplikację napisano w języku C++ i uruchomiono przy pomocy oprogramowania Microsoft Visual Studio .NET 2003.

5. Selekcja pomiarów

Analiza kodu felgi, niezależnie od metody jego identyfikacji, rozpoczyna się od weryfikacji i selekcji pomiarów przesłanych przez mikrokontroler AVR ATmega88 do komputera PC. Poprawne skanowanie dostarcza ok. 5500 wyników pomiarów do analizy. Jeśli ich liczba jest mniejsza niż 1200, to przyjmuje się, że podczas skanowania wystąpił błąd i na podstawie wykonanego skanowania kodu nie da się zidentyfikować. Następnie odrzuca się:

- ok. 2 % wyników pomiarów zarejestrowanych na początku zakresu pomiarowego,
- ok. 25 % wyników pomiarów zarejestrowanych na końcu zakresu pomiarowego – jeśli dalsza część analizy będzie dotyczyła FelgenCode w wersji V1.3,
- ok. 5 % wyników pomiarów zarejestrowanych na końcu zakresu pomiarowego – jeśli dalsza część analizy będzie dotyczyła FelgenCode w wersji V3.0.

Zakresy te ustalono doświadczalnie. Różnica liczby odrzucanych pomiarów z końca zakresu pomiarowego w przypadku wersji V1.3 i V3.0 wynika m.in. z tego, że pole kodowe w wersji V3.0 jest o 25 mm dłuższe, a zakres ruchu sondy jest w obu przypadkach jednakowy.

W wyniku odrzucenia części skrajnych pomiarów można przyjąć, że pozostałe z nich zarejestrowano podczas ruchu głowicy ze stałą prędkością, a więc poza obszarem, gdzie ramię z głowicą jest rozpędzane i hamowane. Aby wyznaczyć bazę odniesienia, czyli średnią odległość głowicy pomiarowej od powierzchni felgi, należy wykorzystać wyniki tych pomiarów, które zarejestrowano podczas ruchu głowicy poza polem kodowym, a więc w położeniach:

- na lewo od skrajnego lewego szerokiego karbu referencyjnego,
- na prawo od skrajnego prawego szerokiego karbu referencyjnego.

Ponieważ dokładne położenie karbów referencyjnych na tym etapie analizy nie jest znane, toteż do wyznaczenia bazy odniesienia bierze się pod uwagę:

- ok. 15 % *początkowych* pomiarów ze zbioru pomiarów przyjętych do dalszej analizy,
- ok. 20 % *końcowych* pomiarów ze zbioru pomiarów przyjętych do dalszej analizy – jeśli identyfikowany ma być kod FelgenCode w wersji V1.3,
- ok. 30 % *końcowych* pomiarów ze zbioru pomiarów przyjętych do dalszej analizy – jeśli identyfikowany ma być kod FelgenCode w wersji V3.0,

Zakresy te także ustalono doświadczalnie i tak je dobrano, aby zachowując pewien margines bezpieczeństwa z dalszej analizy wykluczyć wszystkie pomiary wykonane bezpośrednio nad polem kodowym felgi.

6. Wyznaczanie baz odniesienia

W pierwszych próbach odczytu kodu felgi bazę odniesienia obliczano jako średnią arytmetyczną ze wszystkich *początkowych i końcowych* pomiarów. Rozwiązanie to jest akceptowalne jedynie przy założeniu, że odlew pozostaje idealnie symetryczny i podczas skanowania został tak ustawiony, że jego oś symetrii dokładnie pokrywa się z osią obrotu ramienia z głowicą pomiarową. W praktyce warunki te rzadko są spełnione. Dlatego ostatecznie przyjęto koncepcję wykorzystania w algorytmie odczytu kodu dwóch zmiennych baz odniesienia, których wartości w poszczególnych punktach pomiarowych aplikacja wylicza przyjmując jako argument numer pomiaru.

Pierwsza baza odniesienia, określana jako baza wyznaczona metodą *czterech stref*, ma kształt 4-elementowej linii schodkowej. Jej poszczególne poziomy są obliczane w następujący sposób:

- poziom pierwszy to średnia arytmetyczna ze wszystkich *początkowych* pomiarów przyjętych do dalszej analizy,
- poziom drugi to suma poziomu pierwszego plus 1/3 różnicy między poziomami pierwszym a czwartym,
- poziom trzeci to suma poziomu pierwszego plus 2/3 różnicy między poziomami pierwszym a czwartym,
- poziom czwarty to średnia arytmetyczna ze wszystkich *końcowych* pomiarów przyjętych do dalszej analizy.

Jeśli zbiór pomiarów przyjętych do dalszej analizy zawiera n elementów, to zmiana poziomu bazy odniesienia następuje po $n/4$, $n/2$ i $3n/4$ pomiarach.

Druga baza odniesienia, określana jako baza wyznaczona metodą *najmniejszych kwadratów*, ma kształt prostej o równaniu:

$$f(x) = a \cdot x + b$$

Aby wyznaczyć optymalne współczynniki a , b prostej posłużono się metodą najmniejszych kwadratów. Polega ona na minimalizacji sumy kwadratów różnic między rzeczywistymi pomiarami odczytanymi w kolejnych punktach a ich wartościami teoretycznymi w tych punktach, obliczonymi przy pomocy funkcji $f(x)$:

$$Q(a, b) = \sum_{i=1}^k (Y_i - f(x_i))^2 = \sum_{i=1}^k (Y_i - a \cdot X_i - b)^2$$

gdzie:

k – to liczba pomiarów, na podstawie których program wyznacza bazę (suma liczby *początkowych* i liczby *końcowych* pomiarów ze zbioru pomiarów przyjętych do dalszej analizy).

Y_i – to wartość pomiaru w punkcie X_i ,

X_i – to i -ty punkt pomiarowy. W obliczeniach numerycznych za wartość współrzędnych X_i aplikacja przyjmuje numery punktów, które wykorzystano do wyznaczania bazy odniesienia. Numeracja obejmuje wszystkie pomiary wykorzystywane do odczytu

kodu, a więc także te, które wykonano nad polem kodowym felgi, i których wartości nie są uwzględniane przy wyznaczaniu bazy odniesienia.

Funkcja $Q(a, b)$ przyjmuje wartość minimalną wtedy, gdy jej pochodne liczone względem parametrów a i b są równe 0, tzn.

$$Q'(a) = \frac{dQ(a, b)}{da} = -2 \cdot \sum_{i=1}^k (Y_i - a \cdot X_i - b) = 0$$

$$Q'(b) = \frac{dQ(a, b)}{db} = -2 \cdot \sum_{i=1}^k ((Y_i - a \cdot X_i - b) \cdot X_i) = 0$$

Pochodne te tworzą układ dwóch równań z dwiema niewiadomymi a i b :

$$a \cdot \sum_{i=1}^k X_i + k \cdot b = \sum_{i=1}^k Y_i$$

$$a \cdot \sum_{i=1}^k X_i^2 + b \cdot \sum_{i=1}^k X_i = \sum_{i=1}^k X_i \cdot Y_i$$

Rozwiązaniem tego układu jest:

$$a = \frac{k \cdot \sum_{i=1}^k X_i \cdot Y_i - \left(\sum_{i=1}^k X_i \right) \cdot \left(\sum_{i=1}^k Y_i \right)}{k \cdot \sum_{i=1}^k X_i^2 - \left(\sum_{i=1}^k X_i \right)^2}$$

$$b = \frac{\sum_{i=1}^k Y_i - a \cdot \sum_{i=1}^k X_i}{k}$$

Wyznaczając następujące współczynniki:

$$A = \sum_{i=1}^k X_i \quad B = \sum_{i=1}^k Y_i \quad C = \sum_{i=1}^k X_i^2 \quad D = \sum_{i=1}^k X_i \cdot Y_i$$

otrzymuje się:

$$a = \frac{k \cdot D - A \cdot B}{k \cdot C - (A)^2}$$

$$b = \frac{B - a \cdot A}{k}$$

7. Binarizacja pomiarów

Kolejnym etapem algorytmu identyfikacji kodu jest binaryzacja pomiarów, czyli zastąpienie tablicy wartości kolejnych pomiarów tablicą, której elementy mają wartość albo zero albo jeden (dalej tablica ta będzie określana jako

BitowyObrazKodu[]). Wykonanie binaryzacji polega na wyznaczeniu dla każdego punktu pomiarowego, czyli dla każdego elementu tablicy pomiarów, różnicy między odczytaną wartością pomiaru a wartością bazy odniesienia w tym punkcie. Różnica ta określa wysokość „nierówności” felgi w danym punkcie względem jej powierzchni bocznej. Tak obliczoną różnicę aplikacja porównuje z parametrem opisującym wysokość karbu. Jeśli różnica jest większa niż wartość tego parametru, to danemu pomiarowi zostaje przypisana wartość „1”, w przeciwnym przypadku – wartość „0”. Binarizacja pomiarów oraz dalsza analiza jest wykonywana dwukrotnie – dla bazy odniesienia wyznaczonej metodą *czterech stref* i dla bazy wyznaczonej metodą *najmniejszych kwadratów*.

Początkowo za wartość parametru odniesienia przyjmowano połowę wysokości karbu. Rozwiązanie to dobrze sprawdzało się w przypadku kodów, których szczeliny między karbami znajdowały się na poziomie bocznej powierzchni felgi. Ale nie dotyczyło to wszystkich felg, ponieważ w wyniku zużycia form odlewniczych niektóre pola kodowe miały szczeliny „zarysowane” niewiele poniżej wierzchołków karbów. Dlatego identyfikacja kodu felgi jest wykonywana dwuetapowo. Pierwsza próba jest realizowana przy stałej wartości parametru odniesienia równej połowie wysokości karbu. Jeśli próba nie da rezultatu, wartość tego parametru jest zmieniana w pętli programowej od 10 % do 90 % wysokości karbu. Taka dwuetapowa identyfikacja pozwala skrócić czas analizy – przy parametrze odniesienia równym 50 % wysokości karbu prawidłowa identyfikacja kodu ma miejsce w większości przypadków i nie ma potrzeby wykonywania drugiego etapu.

8. Filtrowanie pomiarów

W wyniku binaryzacji pomiarów do dalszej analizy aplikacja wykorzystuje tablicę **BitowyObrazKodu[]** o elementach równych albo „0” albo „1”. Teoretycznie grupy „zer” i „jedynek” w tej tablicy powinny odpowiadać szczelinom i karbom pola kodowego felgi (rys. 6). Ale w praktyce zawartość tablicy musi być poddana jeszcze kilku modyfikacjom.

Formy odlewnicze ulegają naturalnemu zużyciu, w wyniku czego elementy pola kodowego różnią się wymiarami od formatu narzuconego przez firmę Numtec. Ponadto w procesie technologicznym formy są okresowo pokrywane dodatkowym środkiem ułatwiającym wyjmowanie odlewów. Środek ten powoduje zmianę chropowatości powierzchni felgi, a w konsekwencji fluktuacje wskazań sondy pomiarowej. Dodatkowym czynnikiem mającym wpływ na te fluktuacje są drgania układu pomiarowego lokalnie zakłócające odczyty. Dlatego informacja zawarta w tablicy **BitowyObrazKodu[]** musi zostać dodatkowo przefiltrowana. Filtrowanie polega na:

- eliminacji pojedynczych „jedynek” – jeżeli element tablicy o indeksie $[i]$ jest równy „1” a elementy $[i-1]$ i $[i+1]$ są równe „0”, to aplikacja zmienia wartość elementu o indeksie $[i]$ na „0”;

- eliminacji stref „zer” – jeżeli element tablicy o indeksie $[i-1]$ jest równy „1”, a element $[i]$ jest równy „0”, to aplikacja sprawdza jak szeroka jest strefa w której kolejne elementy tablicy mają wartość „0”. Jeśli szerokość strefy nie przekracza wartości programowego parametru **MaksymalnaSzerokoscZakloce-nia**, to przyjmuje się, że wyniki pomiarów wykonanych w tym przedziale mogą być niewiarygodne, a więc nie odpowiadają pomiarom wykonanym nad szczeliną kodu. Dlatego aplikacja odrzuca je, zmieniając na „1” wartości wszystkich elementów tablicy **BitowyObrazKodu[]** należących do tego przedziału. Doświadczalnie ustalono, że optymalną wartością parametru **MaksymalnaSzerokoscZakloce-nia** jest 8.

Sposób filtracji dobrano doświadczalnie na podstawie analizy wyników pomiarów zarejestrowanych podczas skanowania kilku tysięcy odlewów. W wyniku przeprowadzonej filtracji grupy „zer” i „jedynek” w tablicy **BitowyObrazKodu[]** powinny już odpowiadać odpowiednim elementom pola kodowego felgi. Dla zaostrzenia kryteriów identyfikacji przyjęto założenie określające minimalną szerokość strefy „jedynek” i „zer” pozwalających uznać dany fragment tablicy za obraz szerokiego karbu, wąskiego karbu, szczeliny lub miejsca gdzie brak wąskiego karbu oznacza, że odpowiedni bit kodu ma wartość „0”. Jeśli szerokość strefy jest mniejsza niż jej dopuszczalna wartość minimalna, to aplikacja uznaje, że dotychczasowa analiza nie doprowadzi do prawidłowej identyfikacji kodu i próbuje odczytać go inną metodą.

9. Identyfikacja położenia pola kodowego i ustalenie gęstości pomiarów

Identyfikacja położenia pola kodowego wewnątrz tablicy **BitowyObrazKodu[]** polega na ustaleniu wartości indeksów dwóch elementów tej tablicy, z których pierwszy jest początkiem skrajnego lewego szerokiego karbu referencyjnego, a drugi – końcem skrajnego prawego szerokiego karbu referencyjnego. Jeśli tablica ta jest n -elementowa, to zwiększając w pętli programowej indeks od 0 do $(n-1)$ należy określić:

- minimalną wartość indeksu i_{min} , dla którego element **BitowyObrazKodu** $[i_{min} - 1]$ ma wartość „0”, a element **BitowyObrazKodu** $[i_{min}]$ – wartość „1” (początek skrajnego lewego karbu referencyjnego),
- maksymalną wartość indeksu i_{maz} , dla którego element **BitowyObrazKodu** $[i_{maz}]$ ma wartość „1”, a element **BitowyObrazKodu** $[i_{maz} + 1]$ – wartość „0” (koniec skrajnego prawego karbu referencyjnego).

Mając wyznaczone położenie pola kodowego, można ustalić gęstość pomiarów, tzn. liczbę wykonanych pomiarów na milimetr długości pola kodowego. Jest ona równa:

- $[(i_{maz} - i_{min} + 1)/106]$ – dla kodu **FelgenCode** w wersji V1.3,
- $[(i_{maz} - i_{min} + 1)/131]$ – dla kodu **FelgenCode** w wersji V3.0.

Liczby 106 i 131 są szerokością danego pola kodowego wyrażoną w milimetrach.

Ponieważ szerokości karbów i szczelin (wyrażone w milimetrach) zostały ściśle określone przez firmę Numtec (rys. 2), toteż znając gęstość pomiarów można określić:

- średnie wartości szerokości tych elementów wyrażone w liczbie pomiarów,
- orientacyjne położenie początków i środków karbów wewnątrz tablicy **BitowyObrazKodu[]** określone przez odpowiednie wartości indeksów tej tablicy.

10. Identyfikacja wystąpienia poszczególnych karbów kodu

Obecność karbu lub jego brak oznacza, że odpowiadający mu bit kodu ma wartość odpowiednio „1” lub „0”. Aplikacja komputerowa identyfikuje wystąpienie karbu lub jego brak przy pomocy dwóch metod:

Metoda 1

Metoda polega na zwiększaniu indeksu i wskazującego kolejne elementy tablicy **BitowyObrazKodu[i]**, a następnie rejestrowaniu, czy dla aktualnej wartości tego indeksu nastąpiła zmiana taka, że:

BitowyObrazKodu[i-1] jest równy 0,
BitowyObrazKodu[i] jest równy 1.

Jest to poszukiwanie „zbocza narastającego”, będącego początkiem karbu. Jeśli aplikacja wykryje taką zmianę, to na podstawie wartości indeksu oraz wymiarów pola kodowego ustalane jest, któremu bitowi kodu może odpowiadać wykryty karb. Metoda ta jest jednak mało skuteczna w przypadku felg wykonanych za pomocą zużytych form odlewniczych, których kody mają taką wartość, że odpowiadające im karby bezpośrednio sąsiadują ze sobą. W takich przypadkach karby często oddziela od siebie szczelina niepełnowymiarowa, ale wąskie „zarysowanie”, którego obraz w tablicy **BitowyObrazKodu[]** zostaje dodatkowo zatarty wskutek eliminacji stref „zer” podczas filtrowania.

Metoda 2

Metoda polega na wyznaczeniu takich wartości indeksów tablicy **BitowyObrazKodu[]**, które teoretycznie wskazują na elementy odpowiadające pomiarom wykonanym w pobliżu środka karbu lub środka miejsca, gdzie na polu kodowym karb może wystąpić. Po wyznaczeniu takiego indeksu i pozostaje tylko odczytać wartość elementu **BitowyObrazKodu[i]** („0” lub „1”) oraz zweryfikować szerokość strefy „zer” lub „jedynek” wokół niego. Skuteczność tej metody w znacznym stopniu zależy od utrzymania sta-

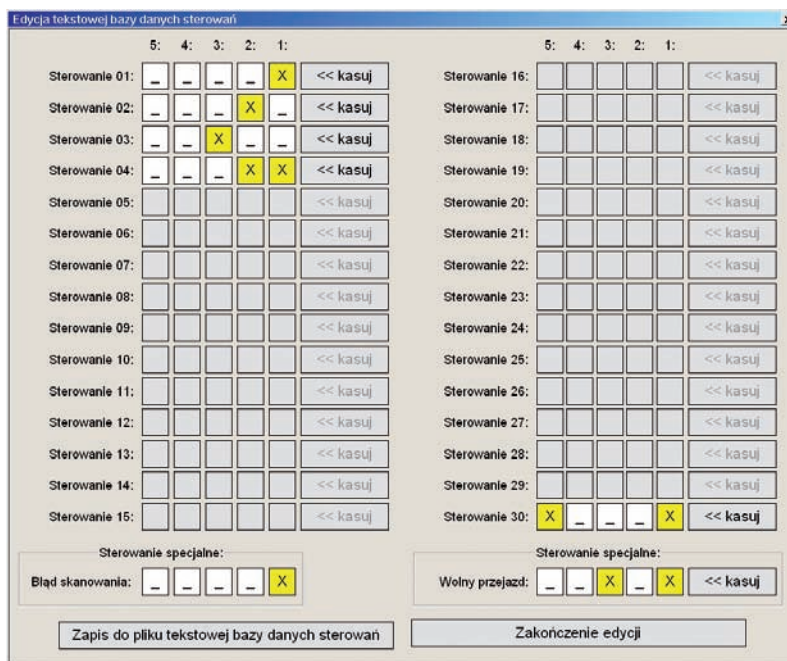
łej prędkości obrotu ramienia z głowicą pomiarową oraz dokładności, z jaką wyznaczono położenie pola kodowego.

11. Wiarygodność i skuteczność odczytu

Przedstawione sposoby identyfikacji kodu felgi wymagają zrealizowania dużej liczby programowych iteracji wykonywanych ze zmieniającymi się wartościami różnych parametrów. Naturalnym pytaniem jest, czy przy pewnych kombinacjach parametrów aplikacja może sygnalizować prawidłowy odczyt, mimo że zidentyfikowany kod nie odpowiada temu, jaki faktycznie znajduje się na powierzchni odlewu. Odpowiedź jest twierdząca, ale takie przypadki są skutecznie eliminowane na drodze dodatkowej weryfikacji:

- format FelgenCode w wersjach V1.3 i V3.0 definiuje dodatkowe elementy pola kodowego, które nie są ani karbami referencyjnymi ani nie odpowiadają poszczególnym bitom samego kodu, ale określają sumę kontrolną wyznaczaną na podstawie wartości tych bitów (jest ona różnie liczona w zależności od wersji kodu),
- w trakcie uruchamiania stanowiska użytkownicy zasugerowali wprowadzenie możliwości tworzenia zbioru numerów felg produkowanych podczas danej zmiany (zwykle dotyczyło to ok. 15 rodzajów). Jeśli odczytany numer felgi nie należy do tego zbioru, to mimo prawidłowego odczytu taka felga jest traktowana jako odlew bez zidentyfikowanego kodu.

Oprogramowanie mikrokontrolera AVR ATmega88 od chwili uruchomienia i włączenia stanowiska do linii technologicznej dobrze radziło sobie z wykrywaniem początków pól kodowych odlewów i nie wymagało dodatko-



Rys. 7. Okienko dialogowe do definiowania kombinacji sygnałów dwustanowych do współpracy z linią technologiczną

Fig. 7. Dialog box for defining combinations of output signals for controlling process Line

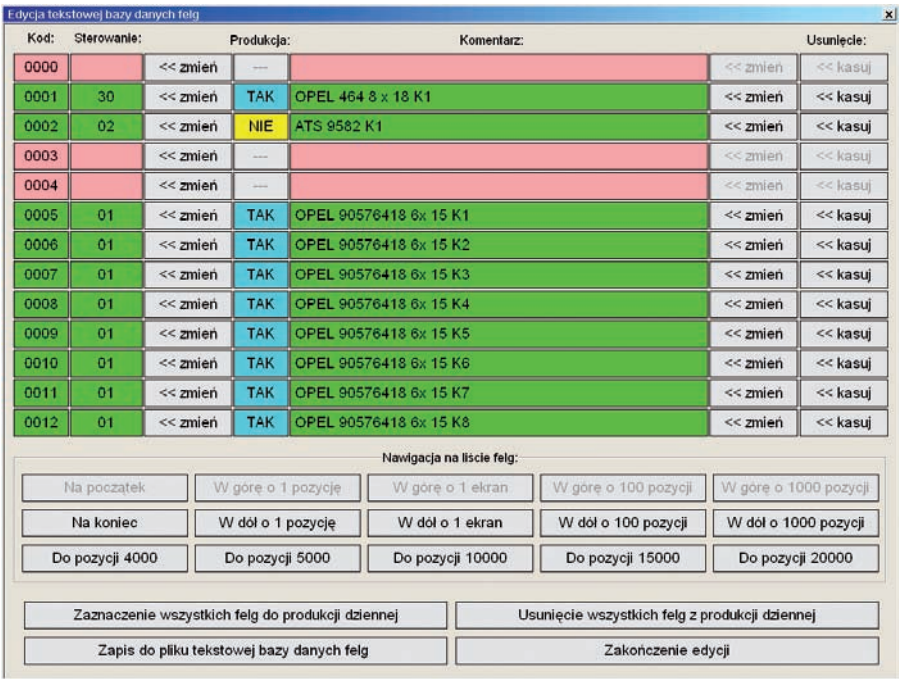
wych poprawek, nawet po wprowadzeniu FelgenCode w wersji V3.0. Podobnie było z programem sterownika Siemens S7-200. Kilkakrotnym modyfikacjom poddawano aplikację komputera PC identyfikującą wartość kodu. Niektóre z modyfikacji były wykonywane jako dodatkowe zlecenia (obsługa kodów w wersji V3.0, sporządzanie statystyk produkcji, blokowanie dostępu do niektórych funkcji przy pomocy hasła itd.), zawsze jednak „przy okazji” doskonalono algorytm odczytu. Także Dział Utrzymania Ruchu Zakładów ATS dążył do poprawiania jakości kodów odlanych na powierzchni felg i wymusił, że na stanowisko odczytu rzadziej trafiają felgi z nadlewkami utrudniającymi ich pozycjonowanie względem osi obrotu ramienia z głowicą pomiarową. Podczas kolejnej modernizacji użytkownicy, wykorzystując własne obserwacje, poprawili sposób mechanicznego pozycjonowania odlewów w trakcie wykonywania pomiarów. Wszystko to doprowadziło do uzyskania skuteczności odczytu oscylującej w granicach 99 %.

12. Współpraca stanowiska odczytu z linią technologiczną

Dodatkowego omówienia wymaga sposób generowania sygnałów o wykrytym kodzie felgi do urządzenia zewnętrznego, w tym przypadku do układu wyboru wykrojnika. W tym celu wykorzystano sześć wyjść dwustanowych z karty WE-WY komputera PC, przy czym pięć z nich przekazuje informację o wyborze wykrojnika, a szósty jest sygnałem potwierdzającym (jest to rozwiązanie mocno nadmiarowe, ponieważ kontrakt na budowę stanowiska przewidywał sterowanie wyborem tylko czterech wykrojników).

Aplikacja komputera PC pozwala na zdefiniowanie do 30 kombinacji tych sygnałów wyjściowych oraz dodatkowych dwóch kombinacji, które sygnalizują błąd skanowania i przejazd bez wyboru wykrojnika. Kombinacje te można przeglądać, edytować i zapamiętywać w tekstowej bazie danych za pomocą (rys. 7) okna dialogowego:

Inne okienko dialogowe (rys. 8) pozwala przyporządkować feldze o wskazanym kodzie numer kombinacji sygnałów wyboru wykrojnika oraz dodatkowy komunikat tekstowy, który jest wyświetlany po wykryciu danej felgi (rys. 9). Korzystając z tego okienka można też włączać lub



Rys. 8. Okienko dialogowe do obsługi bazy danych felg
Fig. 8. Dialog box for car's wheels data base operate

usuwać felgi z asortymentu felg produkowanych podczas danej zmiany produkcyjnej.

Ustawienia wykonane za pomocą okienek dialogowych (rys. 7 i 8) można zapamiętywać w tekstowej bazie danych. Sposób sterowania układem wyboru wykrojnika na podstawie informacji zapamiętanych w plikach tekstowych pozwala uniknąć modyfikacji samej aplikacji w miarę zmiany asortymentu produkcji felg, ponieważ stosowne zmiany należy wprowadzać tylko do odpowiednich baz danych.

13. Inne funkcje realizowane przez aplikację

Stanowisko identyfikacji kodów felg zamontowano na jedynym transporterze między odlewnią a dalszą częścią linii technologicznej. Są więc na nim skanowane wszystkie wyprodukowane odlewy za wyjątkiem tych, które uprzednio nie przeszły testów związanych z badaniem ich struktury promieniami rentgenowskimi. Miejsce to jest dobrym punktem do zbierania informacji o statystyce produkcji pod warunkiem, że skaner działa niezawodnie, a niezawodność tę operatorzy mogą łatwo zweryfikować.

Aplikacja już od pierwszej wersji umożliwiała niezależną rejestrację w bazie danych MS Access oraz w pliku tekstowym odczytanych kodów felg oraz kodów błędów skanowania. Celem tej rejestracji było zbieranie informacji na temat najczęściej występujących błędów odczytu dla ich eliminowania z realizowanego algorytmu. Z czasem okazało się, że funkcja ta może służyć jako wskaźnik poprawnego działania samego stanowiska. Pogorszenie



Rys. 9. Główne okno aplikacji do odczytu kodów felg
Fig. 9. Main dialog box of PC application for scanning car's wheels codes

skuteczności skanowania wskazuje, że stanowisko wymaga przeglądu mechaniki, oczyszczenia głowicy pomiarowej lub, że pojawiają się felgi o nieczytelnym kodzie. Dlatego przy okazji kolejnej modernizacji oprogramowania dołączono funkcję prezentacji aktualnej statystyki błędów skanowania w głównym oknie dialogowym aplikacji (rys. 9).

W ramach kolejnej modernizacji zrealizowano funkcję umożliwiającą zliczanie wszystkich felg zidentyfikowanych na stanowisku podczas 8-godzinnej zmiany z podziałem na poszczególne numery. Podczas każdej zmiany aplikacja rejestruje wszystkie skanowane felgi, a po jej zakończeniu tworzone jest końcowe zestawienie – ile felg oznaczonych danym kodem zidentyfikowano poprawnie oraz na ilu felgach kod nie został prawidłowo odczytany. Statystykę tę aplikacja zapisuje w postaci arkusza kalkulacyjnego w formacie MS Excel oraz pliku tekstowego na dysku twardym lokalnego komputera. Pliki te, oznaczone nazwą nawiązującą do daty i numeru zmiany produkcyjnej, są dostępne w lokalnej sieci LAN.

Aplikacja komputera PC stanowiska umożliwia także rejestrowanie i przeglądanie informacji dotyczącej pomiarów laserowym czujnikiem odległości oraz badanie skuteczności algorytmów odczytu kodu felgi w zależności od różnych czynników zewnętrznych.

Bibliografia

1. Linnik J.W., Metoda najmniejszych kwadratów i teoria opracowywania informacji, Państwowe Wydawnictwo Naukowe, Warszawa 1962.

2. Głuszkowski T., Arkusze kalkulacyjne – przykłady zastosowań, tricki, kruczki, Wydawnictwo Broker, Łódź 1993.
3. Microsoft Visual C++ 6.0 MFC Library Reference, Microsoft Press 1998.
4. Bates J., Tompkins T., Poznaj Visual C++, Wydawnictwo MIKOM, 1999.
5. Petzold Ch., Programowanie Windows, Wydawnictwo RM, Warszawa 1999.

Scanning and decoding car's wheels codes during production process

Abstract: This article presents station for scanning codes of car's wheels, which was built by Industrial Research Institute for Automation and Measurements PIAP for ATS plant in Stalowa Wola. It includes information about algorithms used for identification and discusses some problems during realization of this work.

Keywords: FelgenCode, laser distance sensor, least square method, PC computer application

Artykuł recenzowany, nadesłany 13.12.2013, przyjęty do druku 28.01.2014.

mgr inż. Jacek Dunaj

W 1980 r. ukończył studia na Wydziale Elektrycznym Politechniki Warszawskiej, od 1985 r. jest zatrudniony w Przemysłowym Instytucie Automatyki i Pomiarów PIAP. Specjalizuje się w programowaniu różnego rodzaju sprzętu: mikroprocesorów, kontrolerów, sterowników i robotów przemysłowych, systemów wizyjnych a także komputerów PC programowanych w języku assemblera i C/C++ w środowisku różnych systemów operacyjnych. Współautor oprogramowania dla kilku urządzeń opracowanych w PIAP, a także wielu wdrożeń przemysłowych, w szczególności wymagających współpracy ze sobą kilku różnych urządzeń automatyki i wykorzystania oprogramowania biurowego (baz danych, arkuszy kalkulacyjnych).

e-mail: jdunaj@piap.pl

