

**GRZEGORZ WASIAK**

**Politechnika Łódzka  
Instytut Automatyki**

## **ALGORYTM PRZEKSZTAŁCANIA SCHEMATÓW BLOKOWYCH W PROGRAMIE AUTOCAD**

Recenzent: **dr hab. inż. Jacek Kabziński, prof. PŁ**

Maszynopis dostarczono: 15.12.2011

*Jako przykład komputerowego wspomaganie projektowania w zakresie szeroko rozumianej automatyki przedstawiono algorytm aplikacji wspomagającej przekształcanie schematów blokowych w środowisku programu AutoCAD. W omawianym algorytmie proponuje się uzupełnienie graficznego odwzorowania poszczególnych elementów używanych w schematach blokowych o atrybuty związane z poszczególnymi wejściami i wyjściami rozpatrywanego elementu, które zawierają nazwy elementów dołączonych do tych wejść i wyjść. Daje to możliwość stworzenia biblioteki elementów wykorzystywanych do rysowania schematów blokowych oraz zbudowania swoistej tablicy połączeń poszczególnych elementów w schemacie.*

### **1. WSTĘP**

**Komputerowe wspomaganie projektowania, CAD** (ang. *Computer Aided Design*) – jest pojęciem określającym zastosowanie sprzętu i oprogramowania komputerowego w projektowaniu technicznym. Popularnym przykładem programu typu CAD jest AutoCAD – produkt firmy Autodesk. Znajduje on zastosowanie między innymi w inżynierii mechanicznej, elektrycznej, budowlanej. We wszystkich tych dziedzinach AutoCAD traktowany jest jako program graficzny uzupełniany co najwyżej bibliotekami elementów ułatwiającymi sam proces rysowania. Podobne zastosowania znaleźć można w zakresie szeroko rozumianej automatyki. Brak jest natomiast aplikacji wykorzystujących połączenie możliwości programu AutoCAD w jego warstwie graficznej z możliwościami obliczeniowymi

współpracującego z nim języka AutoLISP. Niniejszy artykuł pokazuje przykład takich możliwości. Jest nim algorytm aplikacji wspomagającej rysowanie schematu blokowego, a następnie przekształcanie go w kolejnych krokach (zgodnie z algebrą schematów blokowych). Każde z przekształceń zostaje odwzorowane graficznie, a w bazie rysunkowej zapisywane są dodatkowo kolejne wypadkowe transmitancje tych przekształceń. Według autora takie rozwiązanie ma nowatorski charakter w zakresie inżynierskiej praktyki projektowania układów automatyki. Zakłada się u czytelnika podstawową znajomość programu AutoCAD, języka AutoLISP oraz zasad ich współdziałania.

## 2. NARZĘDZIA OMAWIANEGO ALGORYTMU

AutoCAD jest programem, którego elementy rysunkowe typu blok mogą zawierać dodatkowe informacje, wykorzystywane również poza samym programem. Ogólnie blok (wyróżniany unikalną nazwą i traktowany przez polecenia edycyjne jako jeden obiekt) jest złożony z powiązanych ze sobą elementów składowych, przy czym oprócz typowych elementów rysunkowych (również tekstów) w jego skład mogą wchodzić tzw. atrybuty – specyficzne „pojemniki” przechowujące zapisane w nich informacje (tekstowe lub numeryczne). Raz zdefiniowany blok może być wstawiany wielokrotnie w różnych miejscach rysunku, m.in. przy różnych kątach obrotu oraz wartościach związanych z nim atrybutów. Wartości atrybutów dla każdego wystąpienia bloku w rysunku można m.in. eksportować poza program i wykorzystywać w plikach tekstowych czy bazach danych, a ich widoczność na ekranie może być sterowana przez użytkownika.

W omawianym algorytmie proponuje się m.in. uzupełnienie graficznego odwzorowania poszczególnych elementów używanych w schematach blokowych (węzeł rozgałęźny i sumacyjny, blok transmitancyjny oraz gałąź przepływu sygnału) o atrybuty związane z poszczególnymi wejściami i wyjściami rozpatrywanego elementu, które zawierają nazwy elementów dołączonych do tych wejść i wyjść. Daje to możliwość stworzenia biblioteki elementów wykorzystywanych do rysowania schematów blokowych oraz zbudowania swoistej tablicy połączeń poszczególnych elementów w schemacie, pozwalając na ciągłą kontrolę i zapis wykonywanych przekształceń.

Kolejną własnością programu AutoCAD, która zostanie wykorzystana w omawianym algorytmie jest możliwość zapisu rysunku w formacie DXF (ang. *Drawing eXchange Format*). Jest to format pozwalający na wymianę rysunków pomiędzy różnymi programami graficznymi w postaci pliku tekstowego. Plik DXF z formalnego punktu widzenia jest pełną listą opisu elementów bazy rysunkowej programu AutoCAD. Elementami listy DXF są kolejne podlisty zawierające wielkości charakteryzujące poszczególne obiekty rysunkowe, przy czym dla każdego obiektu pierwszą podlistą jest zawsze tzw. para kropkowa – lista złożona z dwóch elementów rozdzielonych kropką – np. (-1 . <Entity name:

**60000cc>**). Pierwszy element tej pary kropkowej jest zawsze liczbą całkowitą określającą rodzaj danej zgodnie z kodem DXF. Natomiast drugim elementem tej pary kropkowej jest wielkość charakteryzująca dany obiekt rysunkowy zgodnie z podanym kodem. W przedstawionym przykładzie **-1** oznacza kod wywołujący nazwę identyfikacyjną obiektu, a **<Entity name: 60000cc>** jest tą właśnie nazwą. Oprócz par kropkowych na liście DXF opisu obiektu występują także listy zwykle zawierające dwa lub więcej elementów, przy czym pierwszy z nich zawsze oznacza kod DXF – np. **(10 10.0 150.0 0.0)**. W tym przykładzie **10** jest kodem tzw. punktu głównego obiektu (jest nim punkt początkowy dla linii, punkt wstawienia dla bloku itp), a **10.0 150.0 0.0** to wartości współrzędnych prostokątnych tego punktu. Dla złożonych obiektów rysunkowych (np. blok z atrybutami czy polilinia z kolejnymi wierzchołkami) lista DXF elementu głównego (np. bloku) zawiera w sobie m.in. podlisty określające własności elementów podrzędnych (składowych) elementu głównego (np. wartości atrybutów).

Ogólnie więc pliki DXF określające aktualny stan rysunku pozwalają na modyfikacje wybranych własności na listach związanych z poszczególnymi obiektami rysunkowymi, np. zmianę współrzędnych punktu wstawienia bloku lub łańcucha tekstowego atrybutu określającego nazwę elementu dołączonego do węzła sumacyjnego. Możliwe jest także wymazanie z pliku niepotrzebnego już obiektu. Tak zmieniony plik DXF może zostać uaktualniony w programie, uwidaczniając dokonane modyfikacje.

Najistotniejszym jednak elementem pozwalającym zbudować i zastosować przedstawiany algorytm jest język programowania AutoLISP, w którego interpreter jest wyposażona każda kolejna wersja programu AutoCAD. AutoLISP jest językiem wysokiego poziomu, dobrze nadającym się do zastosowań graficznych i pozwalającym użytkownikowi na tworzenie własnych funkcji i programów. Przykładem może być funkcja wstawiająca do rysunku blok o określonej postaci zależnej od wskazywanych punktów orientujących kierunek przepływu sygnału. Istotną cechą języka AutoLISP jest m.in. operowanie nad danymi typu **nazwa obiektu rysunkowego** (typ **ENAME**) oraz typu **lista** (typ **LIST**).

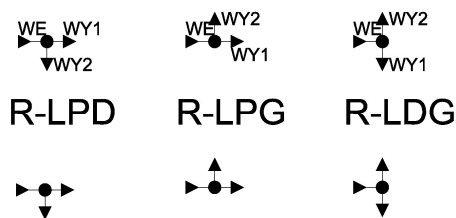
Do obsługi obiektów wykorzystywane mogą być m.in. funkcje: **(entsel)** – pozwala wybrać obiekt rysunkowy, **(entnext)** – odnajduje kolejny (główny lub podrzędny) obiekt w rysunku, **(entget)** – odczytuje listę DXF wybranego obiektu, **(entmod)** – modyfikuje listę DXF wybranego obiektu, **(entdel)** – usuwa obiekt z rysunku, **(entupd)** – aktualizuje wygląd obiektu na rysunku. Natomiast do bezpośrednich działań na listach DXF można zastosować m.in. funkcje: **(car)** – zwraca pierwszy element listy, **(cdr)** – zwraca listę bez pierwszego elementu, **(cons)** – zwraca listę z nowym elementem dodanym na początku listy, **(assoc)** – zwraca element skojarzony z podanym kodem, **(subst)** – zwraca listę po zamianie wybranego elementu. Podane powyżej funkcje występując w programie zapisanym w języku AutoLISP, muszą oczywiście być użyte wraz z niezbędnymi dla nich argumentami, zgodnie z odpowiednią konwencją języka.

### 3. BUDOWA BIBLIOTEKI ELEMENTÓW

Tworzona biblioteka musi być dostępna dla użytkownika poprzez fragment menu programu AutoCAD (np. menu rozwijane, piktogramowe, paskowe lub paletowe), które musi składać się z 4 podstawowych pozycji pozwalających wstawić do rysunku obiekty wykorzystywane przy budowie schematu blokowego. Są nimi w kolejności węzeł rozgałęźny, węzeł sumacyjny, blok transmitancyjny oraz gałąź przepływu sygnału. Każdy z tych obiektów musi zawierać – oprócz elementów rysunkowych składających się na jego graficzne odwzorowanie w programie AutoCAD – także niewidoczne atrybuty pozwalające zapisywać informacje związane m.in. z połączeniami danego obiektu z innymi obiektami w schemacie. Jednocześnie ze względu na możliwą różną orientację danego obiektu na rysunku, należy zaprogramować w języku AutoLISP procedurę wyboru konkretnej postaci obiektu w zależności od wzajemnego położenia punktów, którymi użytkownik odwzorowuje kierunki sygnałów wejściowych i wyjściowych dopływających i wypływających ze wstawianego aktualnie obiektu. Poniżej przedstawione zostaną propozycje budowy bloków dla poszczególnych obiektów i pokazane propozycje procedury wyboru ich określonej postaci.

#### 3.1. Węzeł rozgałęźny

Graficzne możliwości odwzorowania węzła rozgałęźnego zaproponowano na rys. 1, na którym przedstawiono w trzech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. Pokazane warianty (razem z możliwością obrotu wstawianego bloku o kąty  $\pm 90^\circ$  oraz  $180^\circ$ ) wyczerpują możliwości rozgałęziania sygnałów dla przyjętego przepływu tylko w kierunku poziomym oraz pionowym. W każdym z bloków jego punkt wstawienia został umieszczony na początku strzałki oznaczonej atrybutem WE. Atrybut ten (podobnie jak pozostałe atrybuty WY1 i WY2) jest atrybutem niewidocznym, a jego domyślna wartość powinna być ustawiona na 0 – co oznacza brak elementów przyłączonych do danego wejścia/wyjścia. Wartość ta będzie ulegała zmianie po uzupełnieniu rysunku o gałąź przepływu sygnałów.

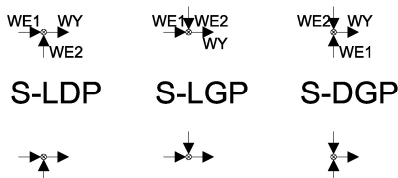


Rys. 1. Bloki odwzorowujące węzeł rozgałęźny

Stworzona w języku AutoLISP procedura wyboru jednego z przedstawionych bloków (wraz z odpowiednim kątem obrotu) powinna być oparta na wskazaniu przez użytkownika – przy włączonym trybie ORTO, co wymusi wskazywanie tylko kierunku poziomego lub pionowego – trzech punktów określających kolejno punkt wstawienia bloku (pokrywający się z kierunkiem sygnału wejściowego WE) oraz kierunki sygnałów WY1 oraz WY2. Dla każdego z sygnałów wyjściowych procedura, opierając się na wartościach współrzędnych zapamiętanych punktów, powinna określić ich położenie względem punktu WE. Przykładowo dla WY1 leżącego poniżej oraz dla WY2 leżącego na lewo od punktu WE, procedura powinna wybrać blok R-LPD z kątem obrotu  $-90^\circ$ .

### 3.2. Węzeł sumacyjny

Graficzne możliwości odwzorowania węzła sumacyjnego zaproponowano na rys. 2, na którym przedstawiono w trzech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. Podobnie jak dla węzła rozgałęźnego pokazane warianty (razem z możliwością obrotu wstawianego bloku o kąty  $\pm 90^\circ$  oraz  $180^\circ$ ) wyczerpują możliwości sumowania sygnałów dla przyjętego przepływu tylko w kierunku poziomym oraz pionowym. W każdym z bloków jego punkt wstawienia został umieszczony na początku strzałki oznaczonej atrybutem WE1. Atrybut ten (podobnie jak pozostałe atrybuty WE2 i WY) jest atrybutem niewidocznym, a jego domyślna wartość powinna być ustawiona na 0 – co oznacza brak elementów przyłączonych do danego wejścia/wyjścia. Wartość ta będzie ulegała zmianie po uzupełnieniu rysunku o gałęzie przepływu sygnałów.

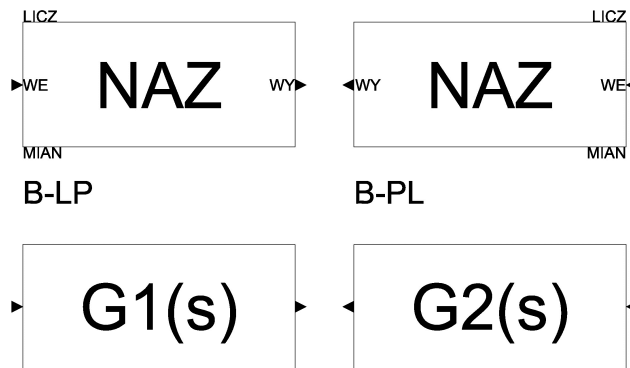


Rys. 2. Bloki odwzorowujące węzeł sumacyjny

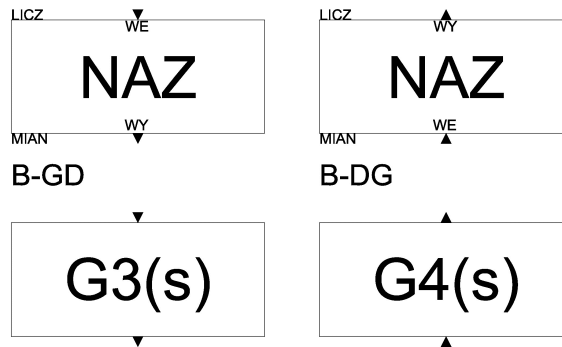
Stworzona w języku AutoLISP procedura wyboru jednego z przedstawionych bloków (wraz z odpowiednim kątem obrotu) powinna być oparta na wskazaniu przez użytkownika – przy włączonym trybie ORTO, co wymusi wskazywanie tylko kierunku poziomego lub pionowego – trzech punktów określających kolejno punkt wstawienia bloku (pokrywający się z kierunkiem sygnału wejściowego WE1) oraz kierunki sygnałów WE2 oraz WY. Dla każdego z sygnałów procedura, opierając się na wartościach współrzędnych zapamiętanych punktów, powinna określić ich położenie względem punktu WE1. Przykładowo dla WE2 leżącego na lewo oraz dla WY leżącego powyżej punktu WE1, procedura powinna wybrać blok S-LGD z kątem obrotu  $+90^\circ$ .

### 3.3. Blok transmitancyjny

Graficzne możliwości odwzorowania bloku transmitancyjnego zaproponowano na rys. 3 oraz rys. 4, na których przedstawiono w czterech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. W odróżnieniu od wcześniej opisanych bloków odwzorowujących węzły schematu blokowego, w definicji bloku transmitancyjnego występuje atrybut NAZ określający nazwę bloku, przy czym (jak widać na rys. 3 i rys. 4) jest on oczywiście atrybutem widocznym po wstawieniu do rysunku. Uniemożliwia to stosowanie do wstawianego bloku różnego od 0 kąta obrotu i wymusza zdefiniowanie dwóch par bloków, których postacie zależą od kierunku sygnałów wejściowego i wyjściowego. Upraszcza to procedurę wyboru bloku, gdyż w tym przypadku należy tylko zbadać wzajemne położenie punktu określającego kierunek sygnału WY względem punktu wstawienia bloku (pokrywającego się z kierunkiem sygnału WE).



Rys. 3. Bloki odwzorowujące bloki transmitancyjne dla poziomego przepływu sygnału



Rys. 4. Bloki odwzorowujące bloki transmitancyjne dla pionowego przepływu sygnału

Dodatkowo definicja każdego bloku zawiera dwa niewidoczne atrybuty LICZ oraz MIAN. Przy wstawianiu bloku zapisuje się w nich współczynniki wielomianów licznika i mianownika opisywanej transmitancji operatorowej. Proponuje się zapis tych współczynników w postaci listy języka AutoLISP.

### 3.4. Gałąź przepływu sygnału

Gałąź przepływu sygnału powinna być wstawiana przy pomocy elementu rysunkowego typu polilinia, charakteryzującego się m.in. możliwością łączenia w jednym obiekcie kilku segmentów liniowych odwzorowujących połączenia pomiędzy wcześniej wstawionymi węzłami i blokami transmitancyjnymi. Procedura rysowania takiej gałęzi powinna umożliwiać (poza standardowym wskazywaniem kolejnych wierzchołków polilinii) zapamiętanie początkowego i końcowego punktu polilinii, przy czym punkt początkowy musi pokrywać się z punktem wyjściowym, a punkt końcowy – z punktem wejściowym odpowiedniego węzła lub bloku transmitancyjnego. Każdy z tych punktów jako argument występującego w języku AutoLISP polecenia (**entsel**) pozwala określić występującą w bazie rysunkowej nazwę obiektu, z którym gałąź jest połączona. Pozwala to zmienić standardowe wartości ustawione na 0 dla odpowiednich atrybutów (wyjściowego dla obiektu przyłączonego do początku i wejściowego dla obiektu przyłączonego do końca gałęzi) na nazwy obiektów przyłączonych do drugiego z krańców gałęzi.

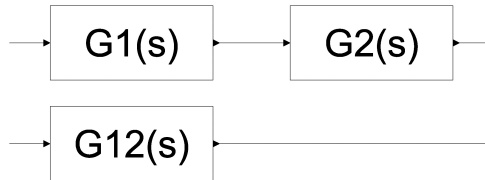
Należy także pamiętać, że wymazywanie z rysunku gałęzi przepływu sygnału musi działać wraz z procedurą odczytującą dla danej gałęzi (poprzez jej punkty krańcowe) nazwy przyłączonych obiektów i zmieniającą wartości ich odpowiednich atrybutów na 0. Takie postępowanie zapewni użytkownikowi, że spośród wszystkich atrybutów typu WE i WY w całej bazie rysunkowej tylko te, które opisują sygnały wejściowe i wyjściowe całego układu będą miały wartości równe 0.

## 4. PRZEKSZTAŁCENIA SCHEMATÓW BLOKOWYCH

Przygotowany w sposób podany powyżej schemat blokowy może zostać poddany przekształceniom wynikającym z algebry schematów. Tworzone procedury, podobnie jak biblioteka elementów, muszą być dostępne dla użytkownika w określonym typie menu. Poniżej zostaną zaproponowane algorytmy procedur wykonujących poszczególne przekształcenia. Należy jednak pamiętać o konieczności modyfikacji każdej z tych procedur w zależności od tego, do jakiej konfiguracji gałęzi przepływu sygnałów jest ona stosowana. Opisane procedury dotyczą konfiguracji przedstawionej na odpowiadających im rysunkach.

#### 4.1. Połączenie szeregowe

Procedura przekształcająca połączenie szeregowe powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w kolejności zgodnej z kierunkiem przepływu sygnału (na rys. 5 oznacza to kolejność  $G1(s) - G2(s)$ ).

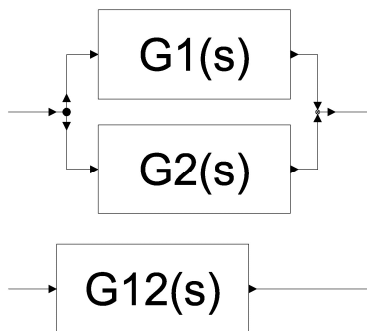


Rys. 5. Połączenie szeregowe – procedura zwijania schematu

Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej sygnał do bloku  $G1(s)$  oraz wyprowadzającej sygnał z bloku  $G2(s)$ . Jednocześnie przesuwa się punkt początkowy gałęzi wychodzącej z  $G2(s)$  do wyjścia bloku  $G12(s)$ . Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku  $G12(s)$ . Na zakończenie usuwane są z rysunku blok  $G2(s)$  oraz gałąź łącząca przekształcane bloki.

#### 4.2. Połączenie równoległe

Procedura przekształcająca połączenie równoległe powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w dowolnej kolejności (na rys. 6 zaproponowano kolejność  $G1(s) - G2(s)$ ).



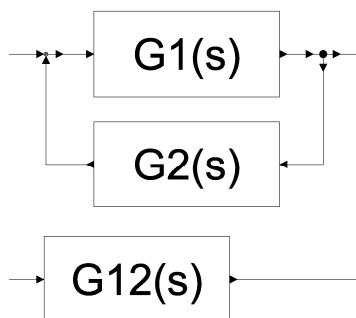
Rys. 6. Połączenie równoległe – procedura zwijania schematu



Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej sygnał do węzła rozgałęźnego oraz wyprowadzającej sygnał z węzła sumacyjnego. Jednocześnie przesuwa się punkt wstawienia bloku  $G12(s)$  do punktu wstawienia węzła rozgałęźnego oraz punkt początkowy gałęzi wychodzącej z węzła sumacyjnego do wyjścia bloku  $G12(s)$ . Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku  $G12(s)$ . Na zakończenie usuwane są z rysunku blok  $G2(s)$ , węzły rozgałęźny i sumacyjny oraz gałęzie łączące te węzły z przekształcanymi blokami.

### 4.3. Połączenie ze sprzężeniem zwrotnym

Procedura przekształcająca połączenie ze sprzężeniem zwrotnym powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w kolejności blok toru głównego oraz blok sprzężenia zwrotnego (na rys. 7 oznacza to kolejność  $G1(s)$  –  $G2(s)$ ).



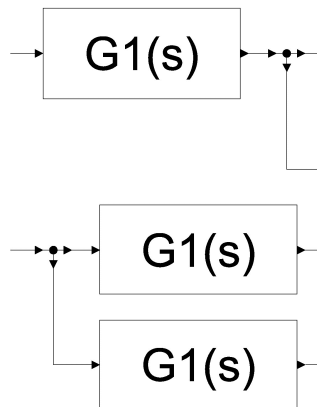
Rys. 7. Połączenie ze sprzężeniem zwrotnym – procedura zwijania schematu

Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej z zewnątrz sygnał do węzła sumacyjnego oraz wyprowadzającej na zewnątrz sygnał z węzła rozgałęźnego. Jednocześnie przesuwa się punkt wstawienia bloku  $G12(s)$  do punktu wstawienia węzła sumacyjnego oraz punkt początkowy gałęzi wychodzącej na zewnątrz z węzła rozgałęźnego do wyjścia bloku  $G12(s)$ . Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku  $G12(s)$ . Na zakończenie usuwane są z rysunku blok  $G2(s)$ , węzły rozgałęźny i sumacyjny oraz gałęzie łączące te węzły z przekształcanymi blokami.

#### 4.4. Przesunięcie węzła rozgałęźnego z wyjścia bloku na jego wejście

Procedura przesuwaną węzeł rozgałęźny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności blok – węzeł (jak na rys. 8).

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła (na rys. 8 jest to wyjście poziome) do wyjścia bloku, a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

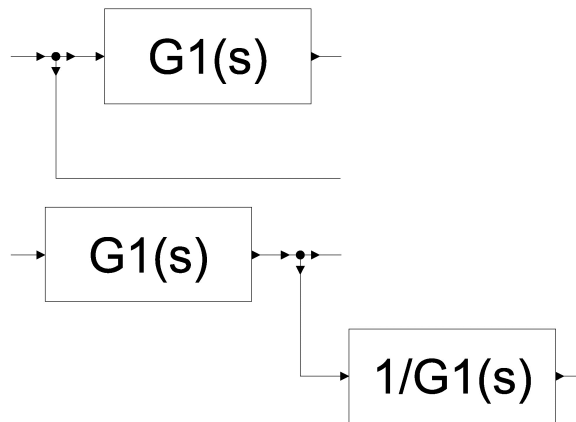


Rys. 8. Przesunięcie węzła rozgałęźnego z wyjścia bloku na jego wejście

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku  $G1(s)$ , łącząc jego punkt wstawienia nową gałęzią z drugim wyjściem węzła rozgałęźnego oraz przesuwaną do wyjścia tego bloku punkt początkowy gałęzi wychodzącej przed przekształceniem z drugiego wyjścia węzła (na rys. 8 jest to wyjście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

#### 4.5. Przesunięcie węzła rozgałęźnego z wejścia bloku na jego wyjście

Procedura przesuwaną węzeł rozgałęźny z wejścia bloku na jego wyjście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności węzeł – blok (jak na rys. 9).



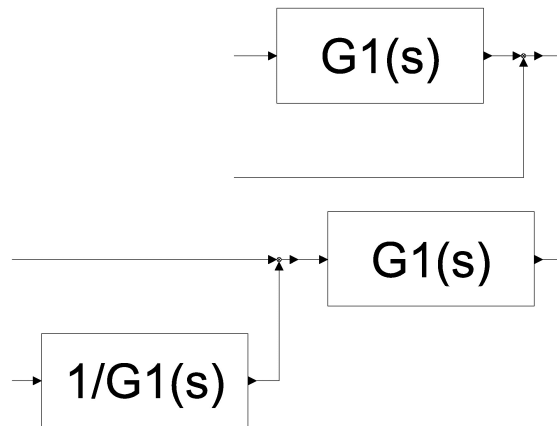
Rys. 9. Przesunięcie węzła rozgałęźnego z wejścia bloku na jego wyjście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z bloku do wyjścia węzła (na rys. 9 jest to wyjście poziome), a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku (o zmienionej nazwie  $1/G1(s)$  i zamienionych wartościach atrybutów LICZ i MIAN), łącząc jego punkt wstawienia nową gałęzią z drugim wyjściem węzła rozgałęźnego oraz przesuując do wyjścia tego bloku punkt początkowy gałęzi wychodzącej przed przekształceniem z drugiego wyjścia węzła (na rys. 9 jest to wyjście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

#### 4.6. Przesunięcie węzła sumacyjnego z wyjścia bloku na jego wejście

Procedura przesuująca węzeł sumacyjny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności blok – węzeł (jak na rys. 10).



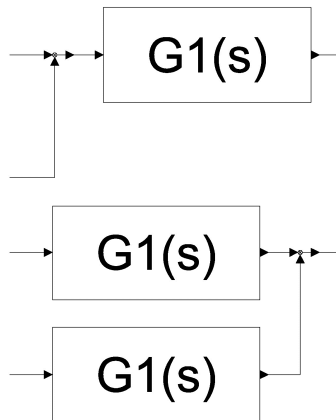
Rys. 10. Przesunięcie węzła sumacyjnego z wyjścia bloku na jego wejście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła do wyjścia bloku, a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła (na rys. 10 jest to wejście poziome). Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku (o zmienionej nazwie  $1/G1(s)$  i zamienionych wartościach atrybutów LICZ i MIAN), łącząc jego punkt wyjściowy nową gałęzią z drugim wejściem węzła sumacyjnego oraz przesuując do wejścia tego bloku punkt końcowy gałęzi dochodzącej przed przekształceniem do drugiego wejścia węzła (na rys. 10 jest to wejście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

#### 4.7. Przesunięcie węzła sumacyjnego z wejścia bloku na jego wyjście

Procedura przesuująca węzeł sumacyjny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności węzeł – blok (jak na rys. 11).



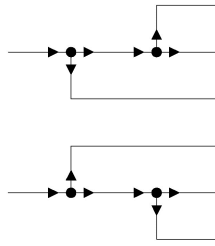
Rys. 11. Przesunięcie węzła sumacyjnego z wejścia bloku na jego wyjście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z bloku do wyjścia węzła, a końcowy punkt gałęzi doprowadzonej do wejścia węzła (na rys. 10 jest to wejście poziome) – do wejścia bloku. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku  $G1(s)$ , łącząc jego punkt wyjściowy nową gałęzią z drugim wejściem węzła sumacyjnego oraz przesuując do wejścia tego bloku punkt końcowy gałęzi dochodzącej przed przekształceniem do drugiego wejścia węzła (na rys. 10 jest to wejście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

#### 4.8. Zamiana węzłów rozgałęźnych

Procedura zamieniająca położenie węzłów rozgałęźnych powinna rozpocząć się od wskazania przez użytkownika dwóch węzłów w kolejności zgodnej z kierunkiem przepływu sygnału.

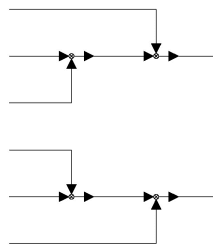


Rys. 12. Zamiana węzłów rozgałęznych

Kolejny krok to usunięcie gałęzi łączącej te węzły, zamiana punktów wstawienia oraz połączenie ich nową gałęzią. Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła drugiego do wyjścia węzła pierwszego (na rys. 12 jest to wyjście poziome), a końcowy punkt gałęzi doprowadzonej do wejścia węzła pierwszego – do wejścia węzła drugiego. Dodatkowo (dla obu węzłów) uwzględnia się przeniesienie punktu początkowego gałęzi przyłączonej do drugiego wyjścia do nowego położenia. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwnych ich krańcach.

#### 4.9. Zamiana węzłów sumacyjnych

Procedura zamieniająca położenie węzłów rozgałęznych powinna rozpocząć się od wskazania przez użytkownika dwóch węzłów w kolejności zgodnej z kierunkiem przepływu sygnału.



Rys. 13. Zamiana węzłów sumacyjnych

Kolejny krok to usunięcie gałęzi łączącej te węzły, zamiana punktów wstawienia oraz połączenie ich nową gałęzią. Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła drugiego do wyjścia węzła pierwszego, a końcowy punkt gałęzi doprowadzonej do wejścia węzła pierwszego – do

wejścia węzła drugiego (na rys. 13 jest to wejście poziome). Dodatkowo (dla obu węzłów) uwzględnia się przeniesienie punktu końcowego gałęzi przyłączonej do drugiego wejścia do nowego położenia. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwnych ich krańcach.

## 5. PODSUMOWANIE

Przedstawiona koncepcja algorytmu jest ogólnym przykładem ilustrującym możliwości wykorzystania programu AutoCAD (rozszerzonego przez aplikację zapisane w języku AutoLISP) w zakresie szeroko rozumianej automatyki. Zbudowanie na tej podstawie aplikacji, która wspomaga przekształcanie schematów blokowych niezależnie od kierunków przepływu sygnałów, wymaga oczywiście dokładnego rozważenia i uwzględnienia wszystkich możliwości konfiguracyjnych schematu. Tym niemniej zaproponowane podejście może być bazą do zaprojektowania całościowego programu.

## LITERATURA

- [1] **Dudek M.:** AutoLISP. Praktyczny kurs. Gliwice. Helion 2003.
- [2] **Pikoń A.:** AutoCAD 2007 PL. Gliwice. Helion 2007.
- [3] Pomoc programu AutoCAD 2009 PL.
- [4] [www.autodesk.pl](http://www.autodesk.pl)

## ALGORITHM FOR THE TRANSFORM OF BLOCK DIAGRAMS IN AutoCAD

### Summary

In the paper, an algorithm is presented for supporting the transformation of block diagrams in the AutoCAD environment, as an example of computer-aided design in the wide range of automatics. The algorithm proposes an extension of a graphical representation of components used in the block diagrams with attributes associated with various inputs and outputs of these components. The attributes contain the names of elements connected to these inputs and outputs. This gives the possibility of creating a library of elements used for drawing block diagrams, and building the specific array of elements' connections in the diagram.

Grzegorz Wasiak  
Instytut Automatyki Politechnika Łódzka