

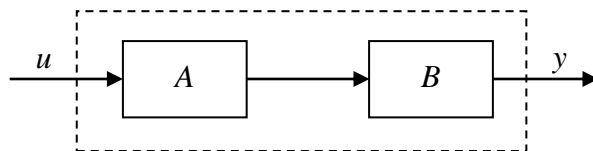
5. Konstrukcja układów dynamicznych

Układ dynamiczny może być zbiorem wielu oddziaływujących na siebie elementów. W zadaniach analizy zachowania liniowych układów stacjonarnych powszechnie stosuje się technikę zastępowania części układu lub nawet całego układu pojedynczym członem zastępczym o takich samych własnościach jak zastępowany układ. Model członu zastępczego definiuje się na podstawie modeli członów składowych zastępowanego układu, wykorzystując informacje o występujących połączeniach (szeregowe, równoległe, sprzężenie zwrotne).

Poniżej omówione zostaną funkcje biblioteki *CST* pozwalające na uzyskanie wypadkowego modelu układu zastępczego z uwzględnieniem występujących w układzie połączeń.

5.1. Połączenie szeregowe, równoległe i sprzężenie zwrotne

5.1.1. Połączenie szeregowe series



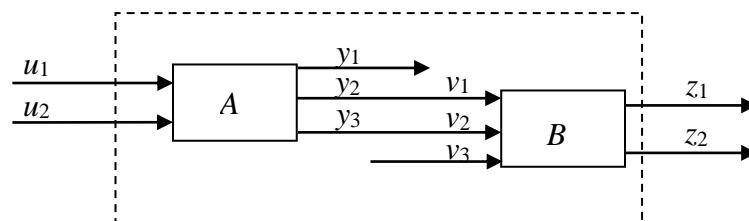
Jeżeli sygnał z wyjścia układu *A* jest podawany na wejście układu *B* to układ zastępczy *C* można utworzyć wykorzystując funkcję **series**. Wywołując funkcję należy podać zmienne reprezentujące układy składowe, funkcja ta zwraca zmienną reprezentującą układ wypadkowy:

```
>> C = series(A, B)
```

Funkcja **series** może być wykorzystana do konstrukcji bardziej ogólnego połączenia szeregowego. W wywołaniu funkcji można podać dodatkowo wektory zawierające odpowiednio numery wyjść układu pierwszego i numery łączonych z nimi wejść układu drugiego.

```
>> C = series(A, B, wyjA, wejB)
```

Na poniższym rysunku przedstawiony został przykładowy sposób połączenia układów *A* i *B*.



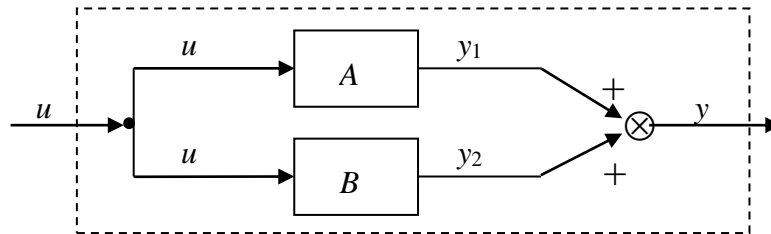
Wyjścia: drugie i trzecie układu *A* są podłączone do pierwszego i drugiego wejścia układu *B*. Połączenie takie można zrealizować wykonując polecenie:

```
>> C = series(A, B, [2 3], [1 2])
```



Po wykonaniu takiego połączenia wejściami wypadkowego układu są wejścia układu A a wyjściami wyjścia układu B .

5.1.2. Połączenie równoległe **parallel**



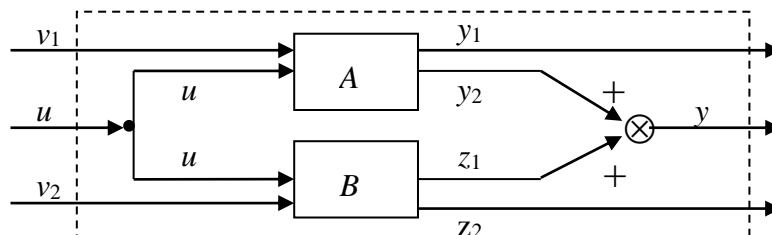
Jeżeli na wejścia układów A i B podawany jest ten sam sygnał, a sygnały wyjściowe układów są sumowane to układ zastępczy C można utworzyć wykorzystując funkcję **parallel**. Wywołując funkcję należy podać zmienne reprezentujące układy składowe, funkcja ta zwraca zmienną reprezentującą układ wypadkowy:

```
>> C = parallel(A, B)
```

Funkcja **parallel** może być wykorzystana do konstrukcji bardziej ogólnego połączenia równoległego. W wywołaniu funkcji można podać dodatkowo wektory zawierające odpowiednio numery łączonych ze sobą wejść układów A i B oraz numery wyjść układu A , które będą zsumowane z podanymi wyjściami układu B .

```
>> C = series(A, B, wejA, wejB, wyjA, wyjB)
```

Na poniższym rysunku przedstawiony został przykładowy sposób połączenia układów A i B .

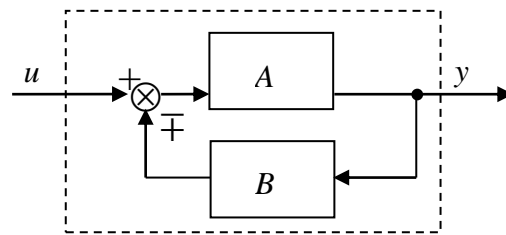


Wejście drugie układu A jest połączone z wejściem pierwszym układu B , na wyjściu układu sumowane są drugie wyjście układu A z pierwszym wyjściem układu B . Połączenie takie można zrealizować wykonując polecenie:

```
>> C = parallel(A, B, [2], [1], [2], [1])
```

Po wykonaniu takiego połączenia wejściami układu C są wejścia v_1 , u , v_2 a wyjściami y_1 , y , z_2 .

5.1.3. Sprzężenie zwrotne **feedback**



Jeżeli sygnał wyjściowy układu A , po dodatkowym przetworzeniu przez układ B , podawany jest przez pętlę *sprzężenia zwrotnego* ponownie na wejście układu A , to układ zastępczy C można utworzyć wykorzystując funkcję **feedback**.

Jeżeli sprzężenie zwrotne jest *sprzężeniem ujemnym* to wywołując funkcję można podać wyłącznie zmienne reprezentujące układy składowe, funkcja **feedback** zwróci zmienną reprezentującą układ wypadkowy.

```
>> C = feedback(A, B)
```

Jeżeli układ ma *dodatnie sprzężenie zwrotne* to wywołując funkcję należy oprócz zmiennych reprezentujących układy składowe podać wartość 1 – definiuje ona sprzężenie dodatnie (dla sprzężenia ujemnego można podać wartość -1), funkcja **feedback** zwraca zmienną reprezentującą układ wypadkowy:

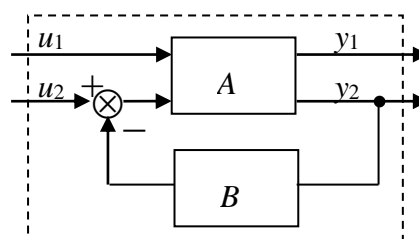
```
>> C = feedback(A, B, 1)
```

Funkcja **feedback** może być wykorzystana do konstrukcji bardziej ogólnego sprzężenia zwrotnego. W wywołaniu funkcji można podać dodatkowo wektory zawierające:

- numery wejść układu z toru głównego które zostaną połączone ze wszystkimi wyjściami układu z toru sprzężenia zwrotnego,
- numery wyjść układu z toru głównego które zostaną połączone ze wszystkimi wejściami układu z toru sprzężenia zwrotnego.

```
C = feedback(A, B, wejA, wyjA)
```

Na poniższym rysunku przedstawiony został przykładowy sposób połączenia układów A i B .



Sygnał z drugiego wyjścia układu A powinien zostać po przetworzeniu przez układ B podany ponownie na drugie wejście układu A . Połączenie takie można zrealizować wykonując polecenie:

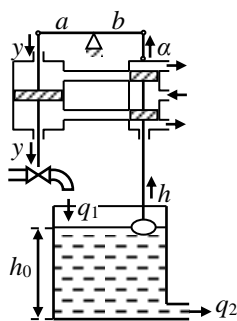


>> **C = feedback(A, B, [2], [2])**

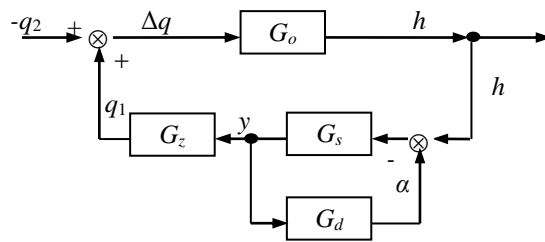
Po wykonaniu takiego połączenia wejściami i wyjściami układu *C* są odpowiednio wejścia i wyjścia układu *A*.

Przykład 5.1.

Należy zbadać zachowanie układu regulacji poziomu cieczy w zbiorniku przedstawionego na rysunku 1. Poziom cieczy w zbiorniku ulega wahaniom z powodu zmian natężenia odpływu q_2 , utrzymanie określonego poziomu h_0 jest możliwe dzięki zmianom natężenia dopływu q_1 .



Rys. 1. Schemat układu



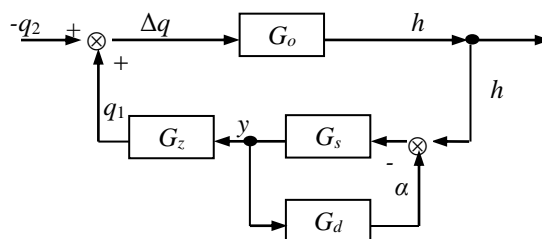
Rys. 2. Schemat blokowy układu

Na rysunku 2. przedstawiony został schemat blokowy układu z rys. 1. Zastosowano następujące oznaczenia dla transmitancji operatorowych elementów tego układu:

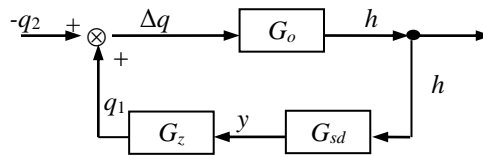
- G_o – transmitancja zbiornika (zbiornik jest obiektem regulacji), $G_o(s) = 1/As$,
- G_s – transmitancja siłownika hydraulicznego, $G_s(s) = 1/Ts$,
- G_d – transmitancja dźwigni, $G_d = b/a$,
- G_z – transmitancja zaworu, $G_z = -k$.

gdzie: A – pole powierzchni podstawy zbiornika; T – stała siłownika; a, b - ramiona dźwigni; k – współczynnik zależny od średnicy zaworu i prędkości przepływu cieczy.

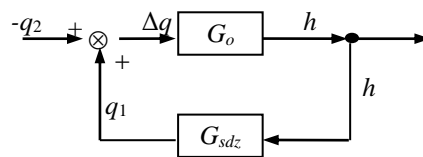
Aby zdefiniować rozważany układ regulacji należy najpierw zidentyfikować połączenia występujące na schemacie. Sygnał z wyjścia siłownika podawany jest, po przekształceniu go przez dźwignię, z powrotem na wejście siłownika ze znakiem „-” jest to więc *ujemne sprzężenie zwrotne*.



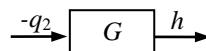
Po zastąpieniu tego elementu układu członem zastępczym o transmitancji G_{sd} , schemat blokowy można przedstawić w postaci:



Sygnal z wyjścia elementu o transmitancji G_{sd} jest podawany na wejście zaworu o transmitancji G_z , połączenie tych elementów jest więc *połączeniem szeregowym*. Wprowadzając człon zastępczy o transmitancji G_{sdz} (zastępujący połączenie szeregowe obiektów G_{sd} i G_z) można ponownie przerysować schemat blokowy układu:



Sygnal z wyjścia obiektu regulacji (tzn. zbiornika) podawany jest, po przekształceniu przez człon o transmitancji G_{sdz} , ponownie na jego wejście ze znakiem „+”, jest to więc *dodatnie sprzężenie zwrotne*. Zastępując sprzężenie zwrotne łączące elementy G_o i G_{sdz} członem zastępczym o transmitancji G można schemat blokowy układu narysować w ostatecznej formie:



Zakładając, że parametry układu wynoszą odpowiednio: $A = 1$ [dm²], $a = b = 1$ [dm], $T = 1$ [s], $k = 10$, a odpływ jest stały w czasie i wynosi $q_2 = 1$ [dm³/s], można napisać skrypt wykonujący symulację działania układu.

```
>> A = 1; k = 10; a = 1; b = 1; T = 1;
```

```
>> Go = tf(1, [A 0]);
```

```
>> Gd = tf(a/b, 1);
```

```
>> Gz = tf(-k, 1);
```

```
>> Gs = tf(1, [T 0]);
```

} definicja elementów składowych układu

— ujemne sprzężenie zwrotne pomiędzy obiektami G_s i G_d

```
>> Gsd = feedback(Gs, Gd,-1);
```

```
>> Gsdz = series(Gsd, Gz);
```

— szeregowe połączenie obiektów G_{sd} i G_z

```
>> G = feedback(Go, Gsdz,1);
```

— dodatnie sprzężenie zwrotne pomiędzy obiektami G_o i G_{sdz}



```
>> t = 0:0.1:10; — czas symulacji  
>> q2 = ones(length(t), 1); — sygnał wejściowy  
>> lsim(G, -q2, t); — symulacja
```

5.2. Konstrukcja złożonego układu: *append* + *connect*

Funkcja **append** pozwala na zgrupowanie układów składowych w jeden układ. Funkcja ta nie tworzy połączeń pomiędzy układami składowymi, utworzona przez nią grupa jest zbiorem niepowiązanych ze sobą układów. Liczba wejść układu wynikowego odpowiada łącznej liczbie wejść układów składowych a ich porządek wynika z kolejności w jakiej układy składowe zostały przekazane funkcji. Podobnie generowane są wyjścia nowego układu, pierwszymi w kolejności są wyjścia pierwszego układu, itd.

Składnia funkcji:

```
sys = append(sys1, sys2, ..., sysN)
```

gdzie: **sys1, sys2, ..., sysN** to układy składowe, **sys** – układ wynikowy.

Funkcja **connect** pozwala na wskazanie przebiegu połączeń wewnętrznych w obrębie grupy utworzonej wcześniej poleceniem **append**.

Wywołując funkcję należy podać:

1. grupę której połączenia wewnętrzne będą definiowane,
2. definicję połączeń
połączenie definiuje się podając numer łączonego wejścia i numery podłączanych do niego wyjść, jeśli sygnał z któregoś z wyjść jest podawany na wejście ze znakiem przeciwnym należy numer tego wyjścia poprzedzić znakiem (-),
definicja pojedynczego połączenia tworzy jeden wiersz macierzy połączeń,
kolejne połączenia definiuje się w kolejnych wierszach macierzy połączeń, jeśli definicja jednego z połączeń wymaga podania mniejszej liczby podłączanych wyjść niż w przypadku innych połączeń należy uzupełnić wiersz definiujący połączenie odpowiednią ilością zer,
3. numery *wejść* grupy układów, które staną się *wejściami* nowego układu,
4. numery *wyjść* grupy układów, które staną się *wyjściami* nowego układu.

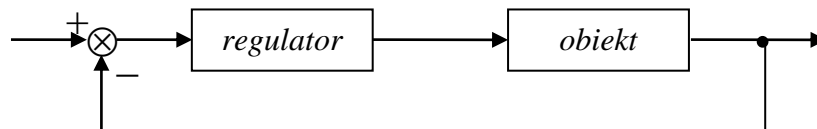
Składnia funkcji:

sys = connect(sysGr, pol, wej, wyj)

gdzie: **sysGr** – grupa układów utworzona przy pomocy funkcji `append`, **pol** - macierz zawierająca definicję połączeń, **wej**, **wyj** - wektory zawierające odpowiednio numery wejść i wyjść tworzonego układu, **sys** – nowy układ, którego struktura odpowiada parametrom zawartym w argumentach: **pol**, **wej** i **wyj**.

Przykład 5.2.1.

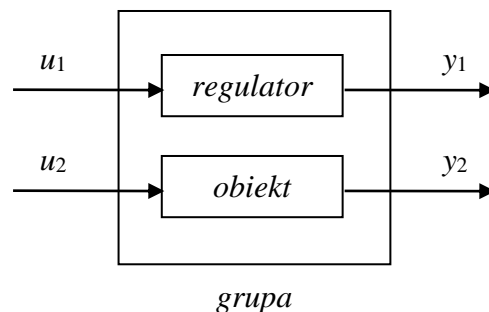
Należy zbudować układ odpowiadający poniższemu schematowi blokowemu:



Po zdefiniowaniu elementów składowych układu, tzn.: *regulatora* i *obiektu*, należy je zgrupować poleceniem **append**:

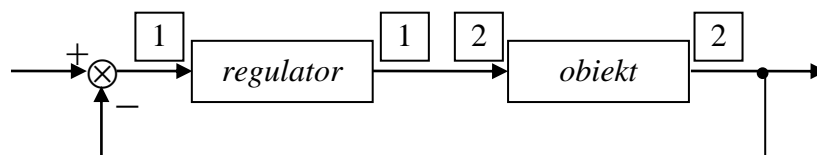
grupa = append(regulator, obiekt)

Wykonanie powyższego polecenia powoduje utworzenie układu o nazwie *grupa*, którego schemat przedstawia poniższy rysunek:



Nowy układ ma dwa wejścia: wejście o numerze 1. jest wejściem *regulatora*, wejście o numerze 2. jest wejściem *obiektu*. Układ ma dwa wyjścia: wyjściem pierwszym jest wyjście *regulatora*, wyjściem drugim wyjście *obiektu*.

Elementy *regulator* i *obiekt* nie są jednak jeszcze ze sobą odpowiednio połączone. Właściwą definicję połączeń ułatwi schemat blokowy układu z naniesioną numeracją wejść i wyjść (zastosowane na schemacie numery wejść i wyjść wynikają z omówionego powyżej sposobu wywołania polecenia **append**).



Z przedstawionego powyżej schematu wynika, że:

- na wejście *regulatora* należy przesłać z wyjścia *obiekta* sygnał o przeciwnym znaku,
- na wejście *obiekta* powinien być przesyłany sygnał z wyjścia *regulatora*.

Dla potrzeb polecenia **connect**, definicję powyższych połączeń można zapisać w postaci macierzy:

[1 -2; 2 1] do wejścia 1. (*regulatora*) podłączany jest sygnał z wyjścia 2. (*obiekta*)
mnożony przez (-1),
do wejścia 2. (*obiekta*) podłączane jest wyjście 1. (*regulatora*)

Ostatecznie, układ odpowiadający schematowi powstanie po wykonaniu polecenia:

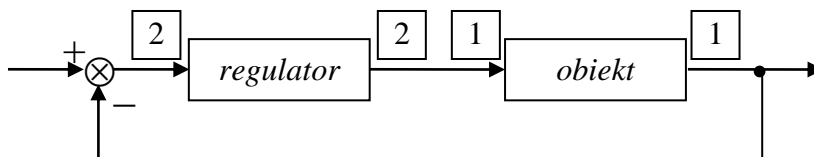
układ = connect(grupa, [1 -2; 2 1], 1, 2)

Dwa końcowe argumenty tego polecenia definiują właściwe wejście i wyjście układu. Wejściem nowego układu jest wejście *regulatora* (wejście nr 1.), wyjściem układu jest wyjście *obiekta* (wyjście nr 2.).

W przypadku gdy do zgrupowania elementów układu użyte zostały polecenie:

grupa = append(obiekt, regulator)

wejście i wyjście *obiekta* otrzymałyby numer 1, a wejście i wyjście *regulatora* numer 2. Uzupełniając schemat układu otrzymaną w ten sposób numeracją

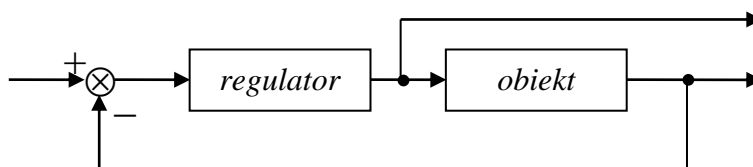


można byłoby ostatecznie utworzyć układ wykonując polecenie:

układ = connect(grupa, [1 2; 2 -1], 2, 1)

Przykład 5.2.2.

Należy zbudować układ odpowiadający poniższemu schematowi blokowemu:



Układ ten różni się od układu z przykładu poprzedniego tylko dodatkowo wyprowadzonym na wyjście sygnałem wyjściowym *regulatora*. W związku z tym, polecenie określające połączenia wewnątrz grupy



będzie się różniło od polecenia z przykładu poprzedniego tylko ostatnim argumentem (argument ten definiuje właśnie wyjścia układu). Wyjściami budowanego układu są wyjścia: *regulatora* i *obiekta*. Ostatecznie układ z powyższego schematu można utworzyć wykonując polecenia:

```
grupa = append(regulator, obiekt)
```

```
uklad = connect(grupa, [1 -2; 2 1], 1, [1 2])
```

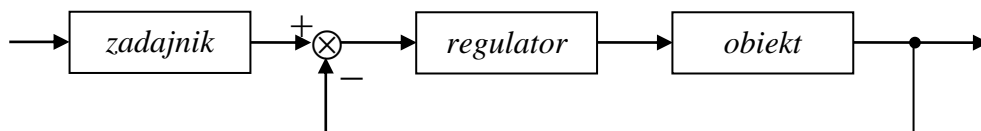
lub

```
grupa = append(obiekt, regulator)
```

```
uklad = connect(grupa, [1 2; 2 -1], 2, [2 1])
```

Przykład 5.2.3.

Należy zbudować układ odpowiadający poniższemu schematowi blokowemu:

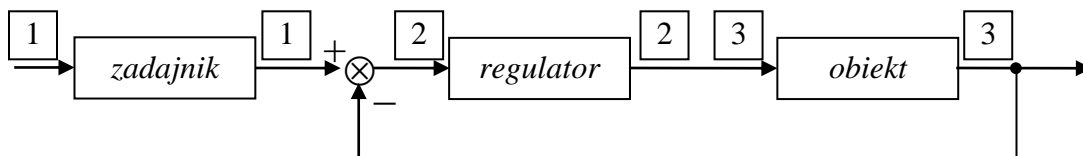


Po zgrupowaniu elementów składowych układu:

```
grupa = append(zadajnik, regulator, obiekt)
```

układ *grupa* będzie miał trzy wejścia i trzy wyjścia. Wejście *zadajnika* będzie pierwszym wejściem tego układu, wejście *regulatora* będzie drugim wejściem, a wejście *obiekta* wejściem trzecim. Na wyjścia układu zostaną kolejno wysłane sygnały wyjściowe *zadajnika*, *regulatora* oraz *obiekta*.

Uwzględniając otrzymaną w ten sposób numerację wejść i wyjść można przerysować schemat układu.



Z przedstawionego schematu wynika, że:

- na wejście *regulatora* należy przesłać sygnał z wyjścia *zadajnika* i z wyjścia *obiekta* sygnał o przeciwnym znaku
- na wejście *obiekta* powinien być przesyłany sygnał z wyjścia *regulatora*.

Definicję powyższych połączeń można zapisać w postaci macierzy:

do wejścia 2. (*regulatora*) podłączany jest sygnał z wyjścia 1. (*zadajnika*) i sygnał z wyjścia 3. (*obiektu*) mnożony przez (-1),

[2 1 -3; 3 2 0] do wejścia 3. (*obiektu*) podłączane jest wyjście 2. (*regulatora*), definicję połączenia **należy uzupełnić zerem**, ponieważ w połączeniu poprzednim na wejście 2. podane zostały sygnały z dwóch wyjść

Wejściem nowego układu jest wejście *zadajnika* (wejście nr 1.), wyjściem układu jest wyjście *obiektu* (wyjście nr 3.). Ostatecznie, układ odpowiadający schematowi powstanie więc po wykonaniu polecenia:

uklad = connect(grupa, [2 1 -3; 3 2 0], 1, 3)

