

POLITECHNIKA POZNAŃSKA
WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI



PRZEDMIOT:
NIELINIOWA TEORIA STEROWANIA – PROJEKT

TEMAT PROJEKTU:
TRACKING CONTROL OF AN UNDERACTUATED SHIP BY
MODIFIED DYNAMIC INVERSION

MAREK FRĄCKOWIAK 144473
SZYMON MURAWSKI 144485

SYSTEMY WIZYJNE SEMESTR 1

PROWADZĄCY:
DR HAB. INŻ. PRZEMYSŁAW HERMAN, PROF. PP

POZNAŃ 2023

Spis treści

1. Wstęp	3
2. Model statku, generator trajektorii, blok uchybów.	3
3. Metoda DEDI.....	7
4. Metoda VIDI.....	11
5. Metoda ORDI.....	14
6. Podsumowanie	17
7. Źródła.....	17

1. Wstęp

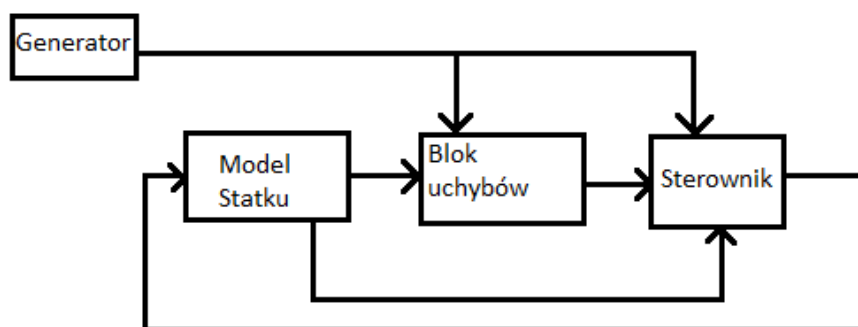
Projekt został opracowany na podstawie artykułu „Tracking control of an underactuated ship by modified dynamic inversion” autorstwa Linqi Ye, Qun Zong. Zadaniem jest implementacja modelu statku i zaproponowanych przez autorów sterowników.

W artykule badany jest problem śledzenia trajektorii niedosterowanego układu (statku) i rozszerzenie klasycznych metod inwersji dynamicznej na obiekty niedosterowane. Przedstawione zostały trzy zmodyfikowane metody inwersji dynamicznej:

- DEDI - Dynamic Extension-Based Dynamic Inversion,
- VIDİ - Virtual Input- Based Dynamic Inversion,
- ORDİ - Output Redefinition- Based Dynamic Inversion.

Wymienione algorytmy sterowania i ich implementacja zostały opisane szczegółowo w rozdziałach 3,4,5. Opisy poszczególnych sygnałów zostały zestawione w tabelach w każdym z rozdziałów dotyczących regulatorów.

Ogólny schemat blokowy układu prezentuje się w następujący sposób.



Rysunek 1 Schemat blokowy układu.

Implementacja schematów regulacji znajduje się w dołączonych do projektu plikach:

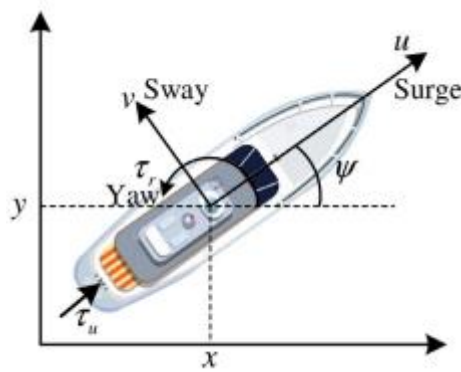
- Sterowanie_statkiemDEDI.slx.
- Sterowanie_statkiemVIDI.slx.
- Sterowanie_statkiemORDI.slx.

W projektach zastosowano stały krok symulacji o rozmiarze 0.001s.

2. Model statku, generator trajektorii, blok uchybów.

Model statku

Statek jest zdefiniowany przy pomocy modelu matematycznego opisanego na dwu wymiarowym układzie współrzędnych. Rozważany model ma 3 stopnie swobody tak jak pokazano na Rysunek 2 Model statku. Zdefiniowane stopnie swobody to Surge(falowanie), Sway (kołysanie), Yaw (ochylenie). Niedosterowanie statku wynika z tego, że w modelu występują tylko 2 sygnały sterujące stopniami swobody (momenty τ_{uu} , τ_{ur}), w układzie nie występuje sygnał sterujący kołysaniem (Sway).



Rysunek 2 Model statku.

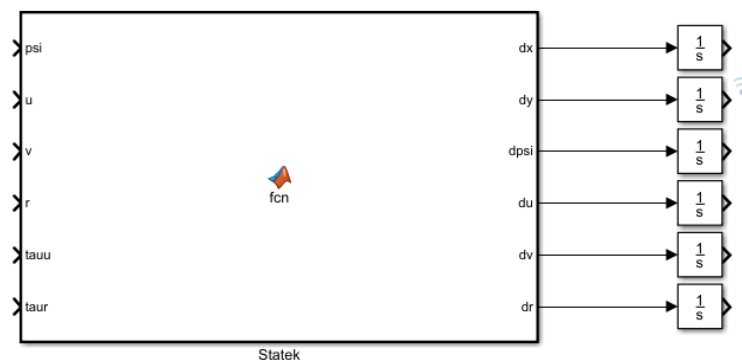
Zgodnie z rysunkiem powyżej, równania dynamiki można zapisać jako:

$$\begin{cases} \dot{x} = u \cos \psi - v \sin \psi \\ \dot{y} = u \sin \psi + v \cos \psi \\ \dot{\psi} = r \\ \dot{u} = (m_{22}vr - d_{11}u + \tau_u) / m_{11} \\ \dot{v} = (-m_{11}ur - d_{22}v) / m_{22} \\ \dot{r} = ((m_{11} - m_{22})uv - d_{33}r + \tau_r) / m_{33} \end{cases} \quad (1)$$

Gdzie:

- x, y – pozycja statku,
- psi – kąt określający orientację statku w układzie,
- u, v, r – prędkości w poszczególnych stopniach swobody (Surge, Sway, Yaw),
- τ_u , τ_r – sygnały sterujące momentem uderu i odchylenia
- m_{ii} , d_{ii} – stałe parametry modelu statku.

Na podstawie podanych równań utworzony został blok funkcyjny w środowisku Simulink. Dodatkowo dodane zostały bloki całkujące co pozwala na zdefiniowanie warunków początkowych podanych w artykule[1] (x = -0.01, y = 1.01, psi = 0.01, u = 0.1, v = 0.1, r = 0.1).



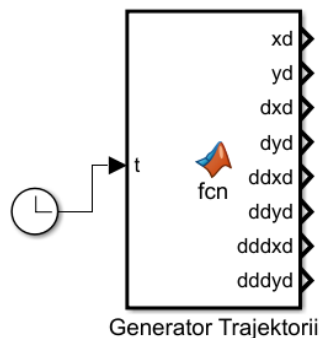
Rysunek 3 Blok funkcyjny statku.

Implementacja równań bloku funkcyjnego statku prezentuje się następująco:

```
function [dx,dy,dpsi,du,dv,dr] = fcn(psi,u,v,r,tauu,taur)
m11=1.956;
m22=2.405;
m33=0.043;
d11=2.436;
d22=12.992;
d33=0.0564;
dx = u*cos(psi)-v*sin(psi);
dy = u*sin(psi)+v*cos(psi);
dpsi = r;
du = (m22*v*r-d11*u+tauu)/m11;
dv = (-m11*u*r-d22*v)/m22;
dr = ((m11-m22)*u*v-d33*r+taur)/m33;
```

Generator trajektorii

Dla potrzeb symulacyjnych generowana jest trajektoria okręgu o punkcie środkowym w miejscu przecięcia osi układu współrzędnych i promieniu 1 m. Wymagana jest również znajomość pochodnych I, II i III rzędu w celu wykorzystania generatora do przeprowadzenia eksperymentów z różnymi metodami sterowania. Zaproponowany generator w środowisku Simulink widoczny jest na rysunku poniżej:



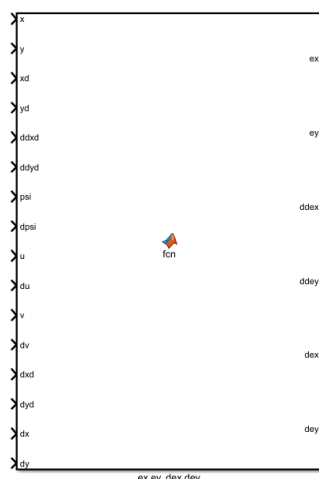
Rysunek 4 Blok funkcyjny generatora.

Aby generator zwracał pożądaną trajektorię opisanego wcześniej okręgu zaimplementowano w nim następujące równania:

```
function [xd,yd,dxd,dyd,ddxd,ddyd,dddxd,dddyd] = fcn(t)
xd = sin(t);
yd = cos(t);
dxd=cos(t);
dyd=-sin(t);
ddxd = -sin(t);
ddyd = -cos(t);
dddxd = -cos(t);
dddyd = sin(t);
```

Blok uchybów

Do poprawnego działania regulatorów wymagane są sygnały uchybów i ich pochodne. W zależności od metody występuje zapotrzebowanie na pochodne różnych rzędów. Do wyznaczenia uchybów i ich pochodnych utworzony został uniwersalny blok realizujący obliczenia.



Rysunek 5 Blok uchybów.

Wyznaczanie pochodnych odbywa się poprzez zaimplementowane w strukturze bloku równania:

```
function [ex,ey,ddex,ddey,dex,dey] =  
fcn(x,y,xd,yd,dxd,ddyd,psi,dpsi,u,du,v,dv,dxd,dyd,dx,dy)  
ex = x-xd;  
ey = y-yd;  
dex=dx-dxd;  
dey=dy-dyd;  
ddex = du*cos(psi)-u*dpsi*sin(psi)-dv*sin(psi)-  
v*dpsi*cos(psi)-ddxd;  
ddey = du*sin(psi)+u*dpsi*cos(psi)+dv*cos(psi)-  
v*dpsi*sin(psi)-ddyd;
```

Uchyby i ich pierwsze pochodne są liczone na podstawie różnicy pomiędzy sygnałami wyjściowymi z bloku statku i sygnałami referencyjnymi z generatora. Druga pochodna uchybu jest zaimplementowana na podstawie równania (4) z artykułu [1].

3. Metoda DEDI

Pierwszą z badanych metod sterowania systemem jest Dynamic Extension-Based Dynamic Inversion (DEDI). Jest ona skierowana dla systemów bez rzędu względnego (relative degree). Aby ją zastosować konieczna jest znajomość pochodnych sygnałów wejściowych. Zostaną one obrane jako nowe, dodatkowe wejścia. Metoda ta ma najpowszechniejsze zastosowania ze wszystkich wymienionych w pracy.

Równanie sterownika, dla interesującego nas obiektu, może być zapisane w następujący sposób:

$$\begin{bmatrix} \xi_u \\ \tau_r \end{bmatrix} = G_1^{-1} \left(-F_1 + \begin{bmatrix} -k_{11}\ddot{e}_x - k_{12}\dot{e}_x - k_{13}e_x \\ -k_{14}\ddot{e}_y - k_{15}\dot{e}_y - k_{16}e_y \end{bmatrix} \right) \quad (2)$$

, gdzie:

$$F_1 = \begin{bmatrix} (a_u - 2\dot{v}r - vf_r - ur^2) \cos \psi - (a_v + 2\dot{u}r + uf_r - vr^2) \sin \psi - \ddot{x}_d \\ (a_u - 2\dot{v}r - vf_r - ur^2) \sin \psi + (a_v + 2\dot{u}r + uf_r - vr^2) \cos \psi - \ddot{y}_d \end{bmatrix} \quad (3)$$

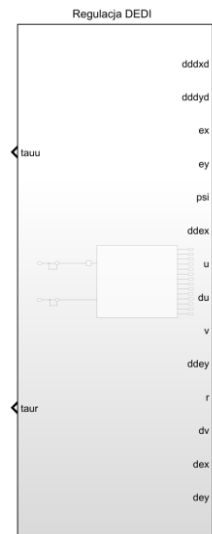
$$G_1 = \begin{bmatrix} \cos \psi / m_{11} & ((m_{22}/m_{11} - 1) v \cos \psi + (m_{11}/m_{22} - 1) u \sin \psi) / m_{33} \\ \sin \psi / m_{11} & ((m_{22}/m_{11} - 1) v \sin \psi + (1 - m_{11}/m_{22}) u \cos \psi) / m_{33} \end{bmatrix}$$

$$f_r = ((m_{11} - m_{22}) uv - d_{33}r) / m_{33} \quad (4)$$

$$\begin{aligned} a_u &= (m_{22}\dot{v}r + m_{22}vf_r - d_{11}\dot{u}) / m_{11}, \\ a_v &= (-m_{11}\dot{u}r - m_{11}uf_r - d_{22}\dot{v}) / m_{22} \end{aligned} \quad (5)$$

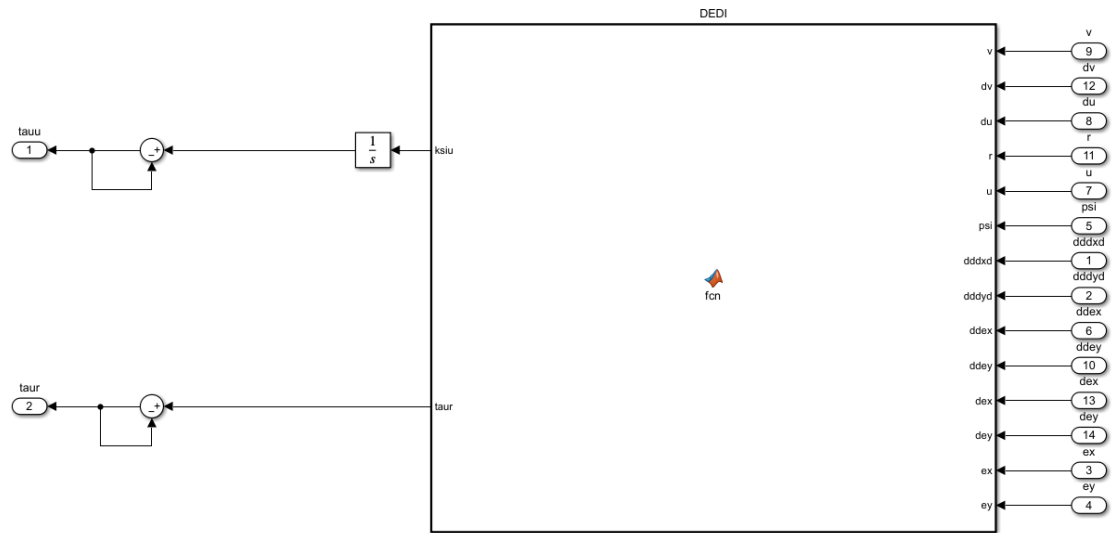
$$\dot{\tau}_u = \xi_u \quad (6)$$

Blok realizujący algorytm DEDI w środowisku Simulink wygląda następująco:



Rysunek 6 Regulator DEDI.

Na poniższym rysunku przedstawiono jego strukturę wewnętrzną:



Rysunek 7 Struktura wewnętrzna regulatora DEDI

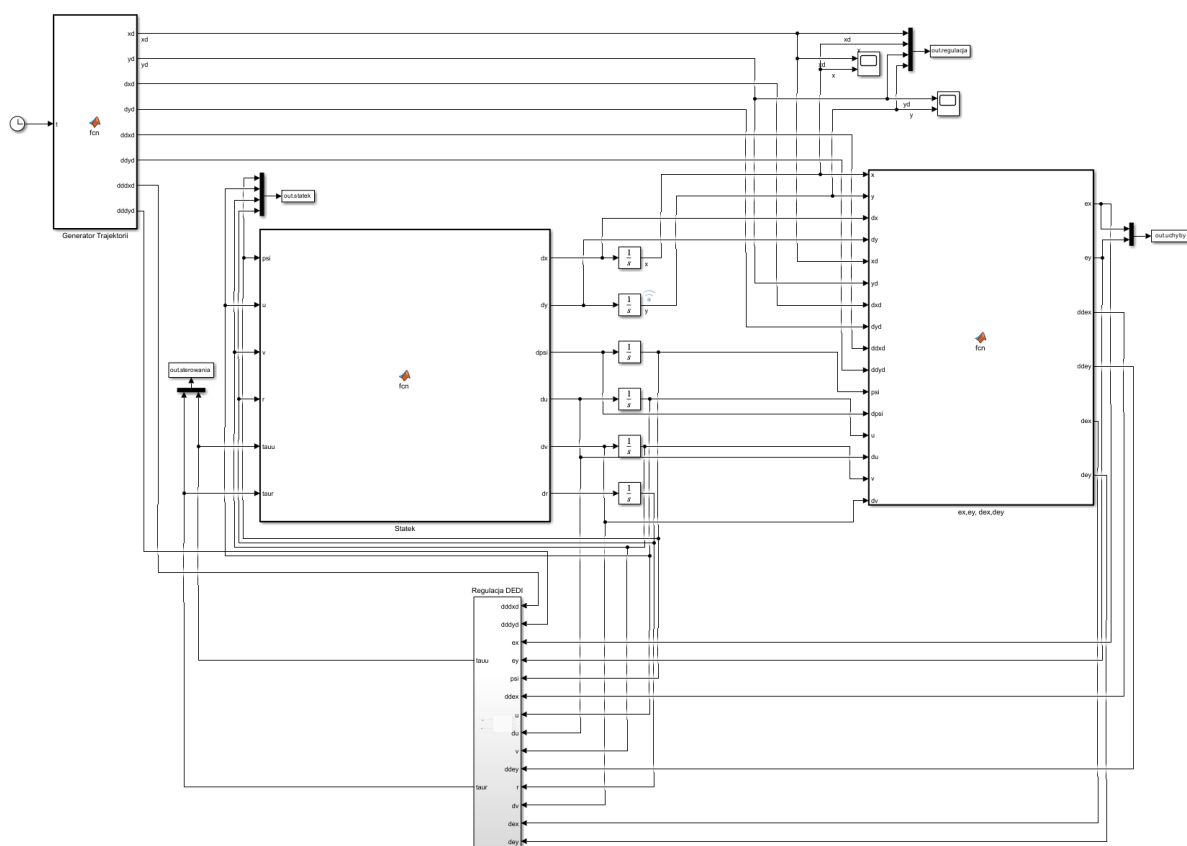
Implementacja bloku funkcyjnego DEDI:

```
function [ksiu,taur] = fnc(v,dv,du,r,u,psi,ddxd,ddy,ddex,ddey,dex,dey,ex,ey)
m11=1.956;
m22=2.405;
m33=0.043;
d11=2.436;
d22=12.992;
d33=0.0564;
k11=6;
k12=12;
k13=8;
k14=6;
k15=12;
k16=8;
fr = ((m11-m22)*u*v-d33*r)/m33;
au = (m22*dv*r+m22*v*fr-d11*du)/m11;
av = (-m11*du*r-m11*u*fr-d22*dv)/m22;
F1 = [(au-2*dv*r-v*fr-u*r^2)*cos(psi)-(av+2*du*r+u*fr-v*r^2)*sin(psi)-ddxd;
(au-2*dv*r-v*fr-u*r^2)*sin(psi)+(av+2*du*r+u*fr-v*r^2)*cos(psi)-ddy];
G1 = [cos(psi)/m11, ((m22/m11-1)*v*cos(psi)+(m11/m22-1)*u*sin(psi))/m33;
sin(psi)/m11, ((m22/m11-1)*v*sin(psi)+(1-m11/m22)*u*cos(psi))/m33];
wek = G1^(-1)*(-1*F1+[-k11*ddex-k12*dex-k13*ex;-k14*ddey-k15*dey-k16*ey]);
ksiu = wek(1);
taur = wek(2);
```


W poniższej tabeli zestawiono nazwy oraz opis wszystkich sygnałów występujących w implementacji tej metody:

Oznaczenie	Opis
dddx	Trzecia pochodna sygnału referencyjnego współrzędna x
dddy	Trzecia pochodna sygnału referencyjnego współrzędna y
ex	Uchyb regulacji dla współrzędnej x
ey	Uchyb regulacji dla współrzędnej y
psi	Kąt kursu statku
ddex	Druga pochodna sygnału uchybu dla współrzędnej x
ddey	Druga pochodna sygnału uchybu dla współrzędnej y
u	Prędkość w surge
du	Pochodna prędkości w surge
v	Prędkość w sway
dv	Pochodna prędkości w sway
r	Prędkość w yaw
dex	Pierwsza pochodna sygnału uchybu dla współrzędnej x
dey	Pierwsza pochodna sygnału uchybu dla współrzędnej y
tauu	Sygnał sterujący, siła w surge
taur	Sygnał sterujący, moment w yaw
ksiu	Pierwsza pochodna siły w surge

Cały schemat regulacji z wcześniej opisanymi blokami został przedstawiony poniżej:

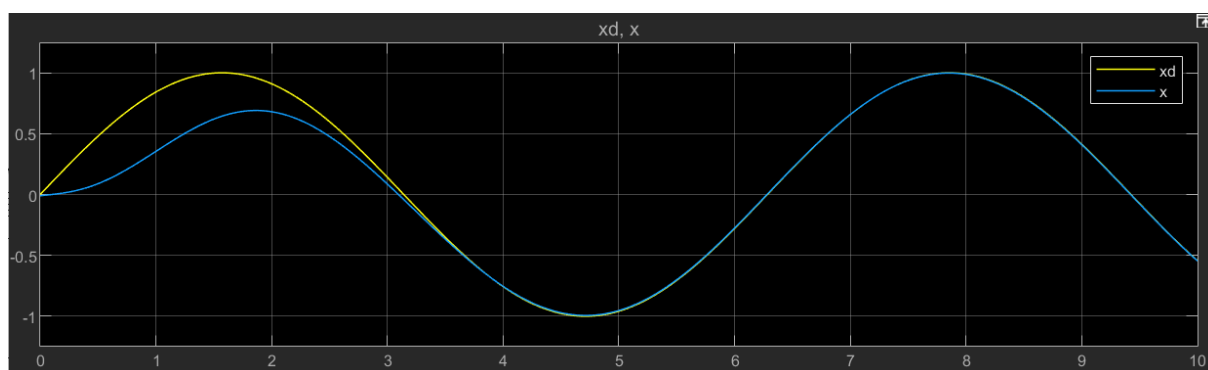


Rysunek 8 Schemat regulacji z regulatorem DEDI.

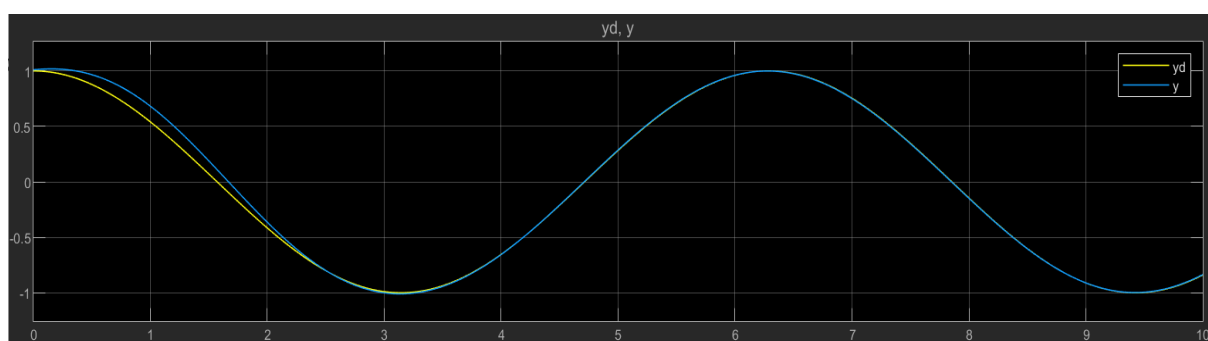
W przygotowanych plikach następuje automatyczny zapis, do przestrzeni roboczej Matlab, ważniejszych sygnałów występujących w projekcie. Nazwy tablic oraz sygnałów w nich się znajdujących zostały przedstawione w tabeli poniżej. Są one identyczne dla każdego z projektów:

Nazwa tablicy	Sygnały		
statek	psi, u, v, r	uchyby	ex, ey
sterowania	taur, tauu		
regulacja	xd, x, yd, y		

Rezultaty eksperymentów:



Rysunek 9 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi X.



Rysunek 10 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi Y.

4. Metoda VIDI

Drugą z metod prezentowanych w artykule jest metoda Virtual Input- Based Dynamic Inversion (VIDI). W algorytmie tym stosuje się wejście wirtualne, które pozwala na uzyskanie rzędu względnego. Metoda ta jest oceniana na łatwiejszą w implementacji, niż metoda DEDI. Wymaga zastosowania inwersji oryginalnej dynamiki. Równania sterownika metody VIDI, zostały przedstawione niżej:

$$\begin{bmatrix} \tau_u \\ \tau_d \end{bmatrix} = G_2^{-1} \left(-F_2 + \begin{bmatrix} -k_{21}\dot{e}_x - k_{22}e_x \\ -k_{23}\dot{e}_y - k_{24}e_y \end{bmatrix} \right) \quad (7)$$

, gdzie:

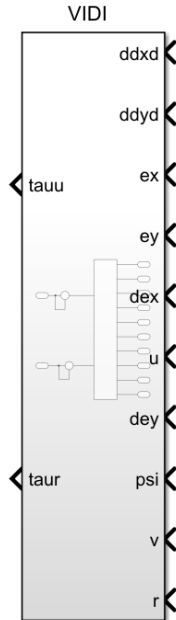
$$F_2 = \begin{bmatrix} -d_{11}u \cos \psi / m_{11} + d_{22}v \sin \psi / m_{22} - \ddot{x}_d \\ -d_{11}u \sin \psi / m_{11} - d_{22}v \cos \psi / m_{22} - \ddot{y}_d \end{bmatrix} \quad (8)$$

$$G_2 = \begin{bmatrix} \cos \psi / m_{11} & m_{22}v \cos \psi / m_{11} + m_{11}u \sin \psi / m_{22} - u \sin \psi - v \cos \psi \\ \sin \psi / m_{11} & m_{22}v \sin \psi / m_{11} - m_{11}u \cos \psi / m_{22} + u \cos \psi - v \sin \psi \end{bmatrix}$$

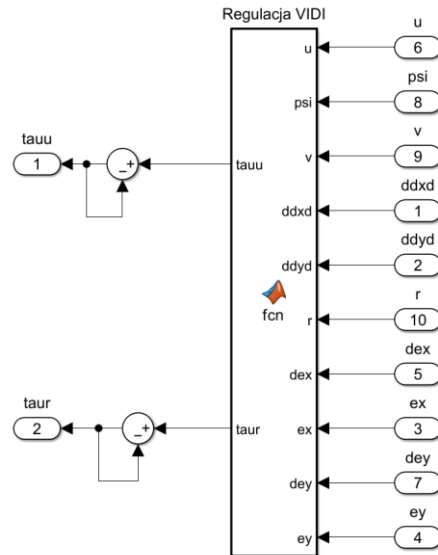
$$\tau_r = g_r^{-1} (\dot{r}_d - f_r - k_{25}e_r) \quad (9)$$

$$f_r = ((m_{11} - m_{22})uv - d_{33}r) / m_{33}, \quad g_r = 1/m_{33} \quad (10)$$

Blok realizujący algorytm VIDI w środowisku Simulink oraz jego struktura wewnętrzna wyglądają następująco:



Rysunek 11 Bok regulatora VIDI.



Rysunek 12 Struktura wewnętrzna regulatora VIDI.

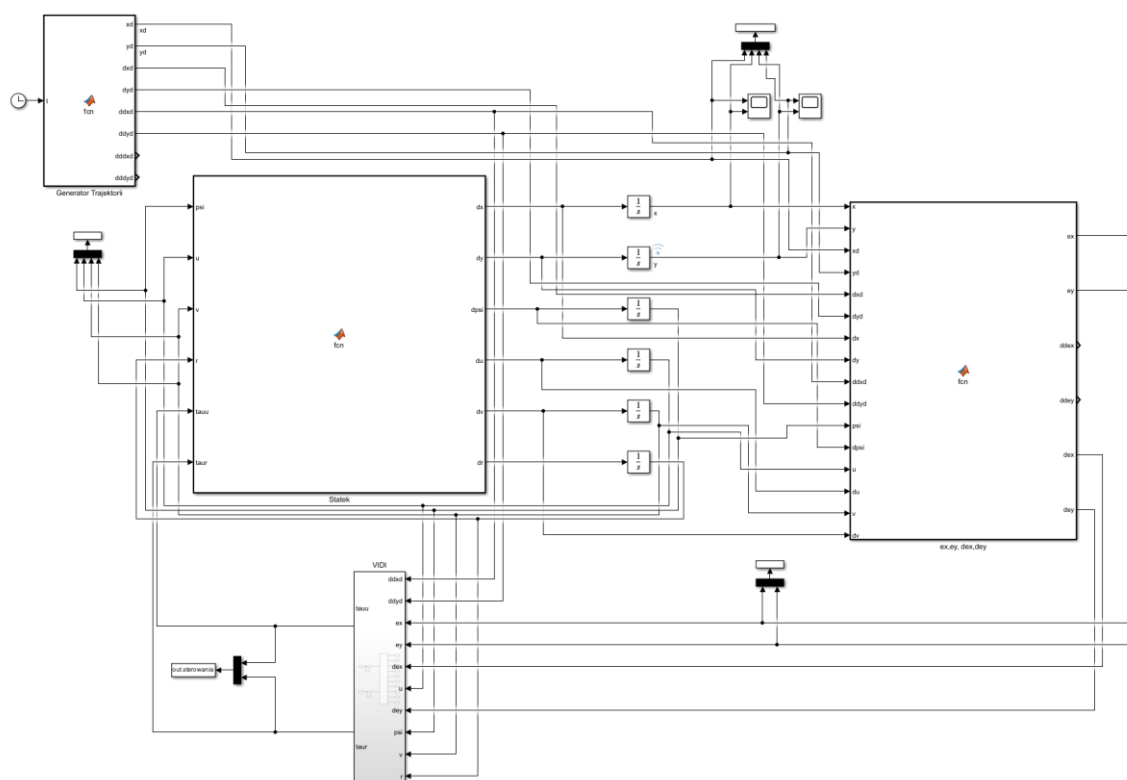
Implementacja bloku funkcyjnego sterowania VIDI:

```
function [tauu, taur] = fcn(u,psi,v,ddxd,ddy,d,r,dex,ex,dey,ey)
k21=10;
k22=10;
k23=10;
k24=10;
k25=2;
lambda=10;
m11=1.956;
m22=2.405;
m33=0.043;
d11=2.436;
d22=12.992;
d33=0.0564;
fr = ((m11-m22)*u*v-d33*r)/m33;
gr = 1/m33;
G2 = [cos(psi)/m11 m22*v*cos(psi)/m11+m11*u*sin(psi)/m22-u*sin(psi)-v*cos(psi);
sin(psi)/m11 m22*v*sin(psi)/m11-m11*u*cos(psi)/m22+u*cos(psi)-v*sin(psi)];
F2 = [-d11*u*cos(psi)/m11+d22*v*sin(psi)*m22-ddxd; -d11*u*sin(psi)/m11-
d22*v*cos(psi)/m22-ddyd];
wek = (G2)^(-1)*(-1*F2+[-k21*dex-k22*ex;-k23*dey-k24*ey]);
tauu = wek(1);
rd = wek(2);
taur = gr^(-1)*(lambda*(rd-r)-fr-k25*(rd-r));
```

W poniższej tabeli zestawiono nazwy oraz opis wszystkich sygnałów, które nie pojawiły się w implementacji metody DEDI, występujących w implementacji tej metody:

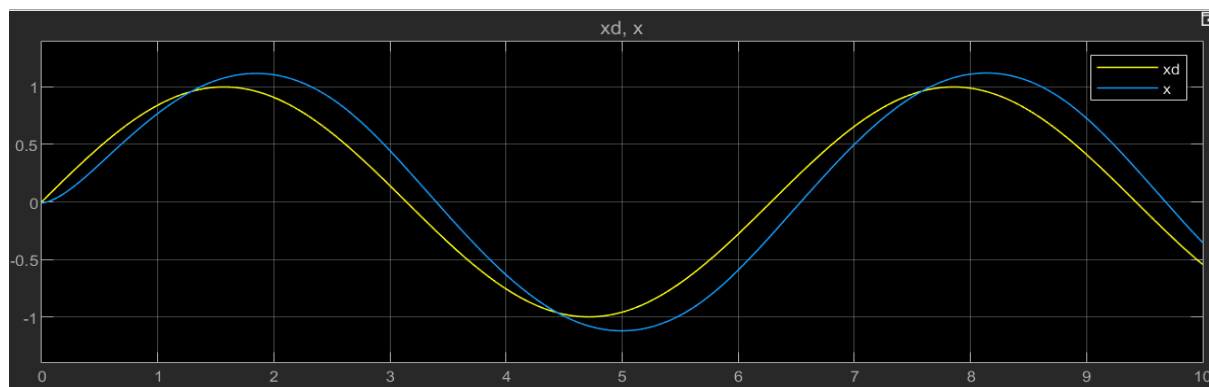
Oznaczenie	Opis
ddxd	Druga pochodna sygnału referencyjnego współrzędna x
ddy	Druga pochodna sygnału referencyjnego współrzędna y

Cały schemat regulacji z wcześniej opisanymi blokami został przedstawiony poniżej:

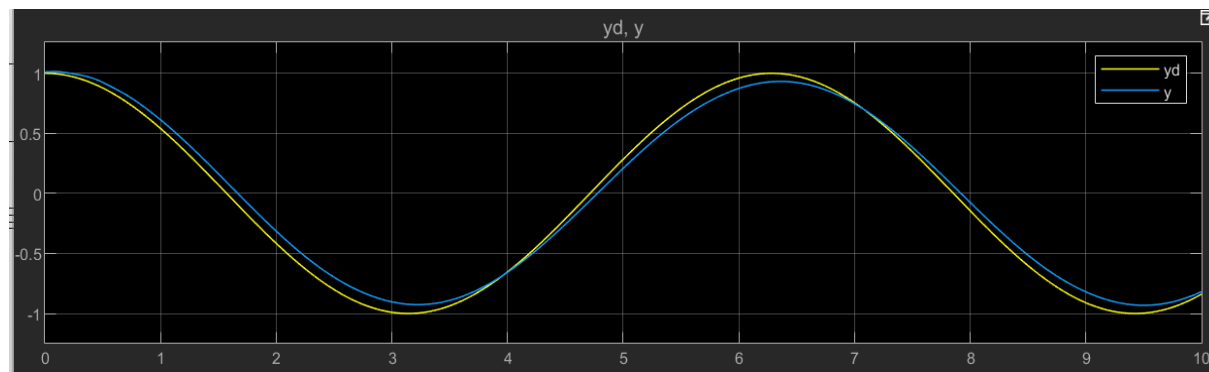


Rysunek 13 Schemat regulacji z regulatorem VIDI.

Wyniki eksperymentów:



Rysunek 14 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi X.



Rysunek 15 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi Y.

5. Metoda ORDI

Trzecią i ostatnią metodą zaproponowaną w opracowywanej publikacji jest Output Redefinition- Based Dynamic Inversion (ORDI). Pierwotnie została zaproponowana dla układów nieminimalnofazowych, jednak okazała się skuteczna również dla układów bez rzędu względnego. Wyjście układu w kombinacji z jedną ze zmiennych stanu jest obierane jako nowe wyjście, co prowadzi model dynamiczny do nowego wejścia. Zadanie śledzenia nie ulega zmianie. Nowe wyjście powinno zostać dobrane starannie, aby błąd śledzenia oryginalnego sygnału, był minimalny. Nowe wyjście powinno być aproksymacją oryginalnego wyjścia. Metoda ORDI może z powodzeniem być stosowana jako algorytm sterujący dla pojazdów hipersonicznych.

Równania sterownika ORDI:

$$\begin{bmatrix} \tau_u \\ \tau_r \end{bmatrix} = G_3^{-1} \left(-F_3 + \begin{bmatrix} -k_{31}\dot{e}_{x_1} - k_{32}e_{x_1} \\ -k_{33}\dot{e}_{y_1} - k_{34}e_{y_1} \end{bmatrix} \right) \quad (11)$$

, gdzie:

$$\begin{aligned} F_3 &= F + \begin{bmatrix} -lf_r \sin \psi - lr^2 \cos \psi \\ lf_r \cos \psi - lr^2 \sin \psi \end{bmatrix} \\ G_3 &= \begin{bmatrix} \cos \psi / m_{11} & -l \sin \psi / m_{33} \\ \sin \psi / m_{11} & l \cos \psi / m_{33} \end{bmatrix} \end{aligned} \quad (12)$$

$$\begin{aligned} x_1 &= x + l \cos \psi \\ y_1 &= y + l \sin \psi \end{aligned} \quad (13)$$

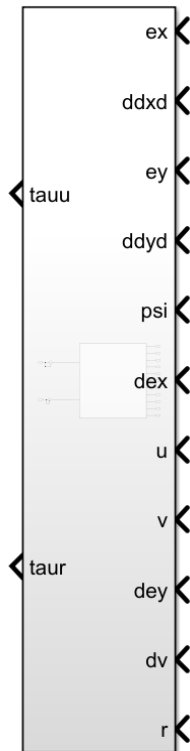
$$\begin{aligned} F &= \begin{bmatrix} f_u \cos \psi - ur \sin \psi - \dot{v} \sin \psi - vr \cos \psi - \ddot{x}_d \\ f_u \sin \psi + ur \cos \psi + \dot{v} \cos \psi - vr \sin \psi - \ddot{y}_d \end{bmatrix} \\ G &= \begin{bmatrix} \cos \psi / m_{11} & 0 \\ \sin \psi / m_{11} & 0 \end{bmatrix} \end{aligned} \quad (14)$$

$$f_u = (m_{22}vr - d_{11}u) / m_{11}. \quad (15)$$

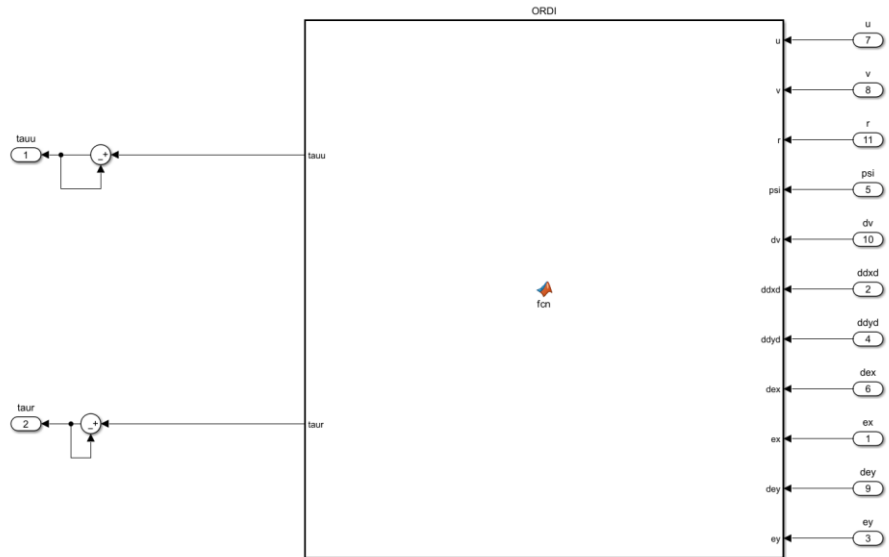
$$f_r = ((m_{11} - m_{22})uv - d_{33}r) / m_{33} \quad (16)$$

Blok realizujący algorytm ORDI w środowisku Simulink oraz jego struktura wewnętrzna wyglądają następująco:

Regulacja ORDI



Rysunek 16 Blok regulatora ORDI.



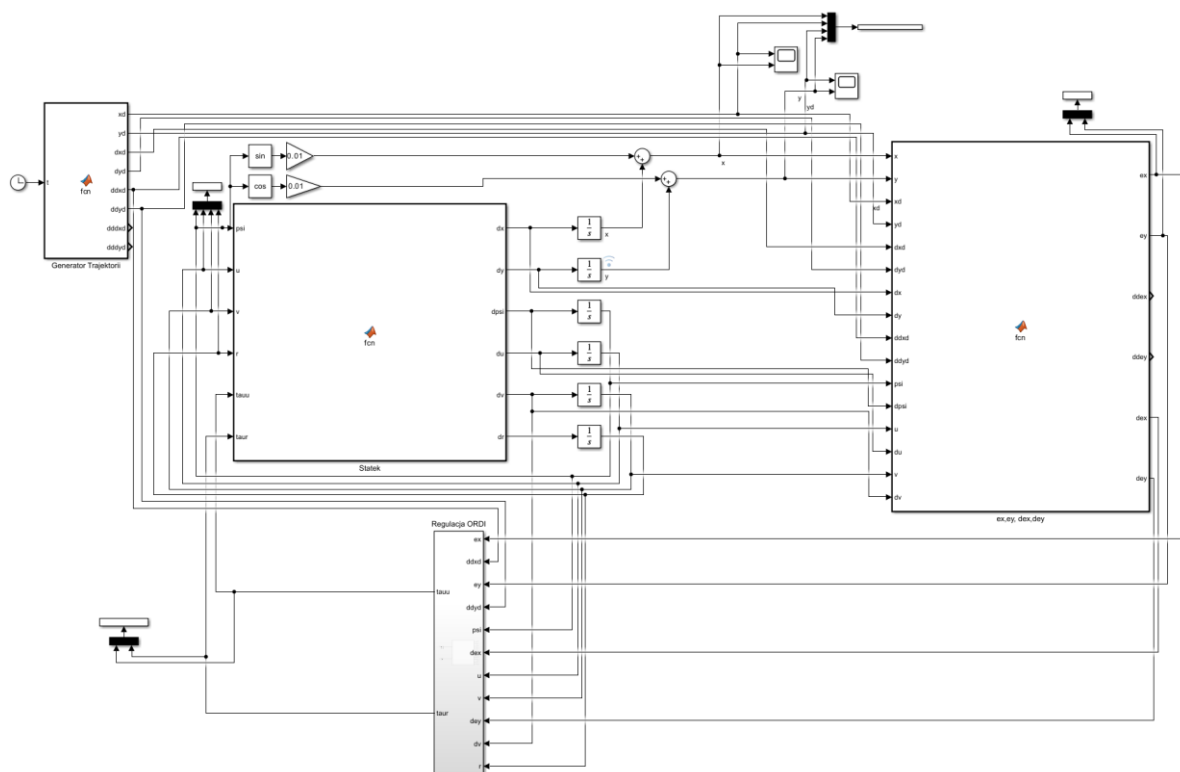
Rysunek 17 Struktura wewnętrzna regulatora ORDI.

Implementacja bloku funkcyjnego sterowania ORDI:

```
function [tauu,taur] = fcn(u,v,r,psi,dv,ddx,ddy,dex,ex,dey,ey)
m11=1.956;
m22=2.405;
m33=0.043;
d11=2.436;
d22=12.992;
d33=0.0564;
k31=4;
k32=4;
k33=4;
k34=4;
l=0.01;
fu = (m22*v*r-d11*u)/m11;
fr = ((m11-m22)*u*v-d33*r)/m33;
F=[fu*cos(psi)-u*r*sin(psi)-dv*sin(psi)-v*r*cos(psi)-ddx;
fu*sin(psi)+u*r*cos(psi)+dv*cos(psi)-v*r*sin(psi)-ddy];
F3 = F + [-l*fr*sin(psi) - l*r^2*cos(psi);l*fr*cos(psi)-l*r^2*sin(psi)];
G3 = [cos(psi)/m11, -l*sin(psi)/m33; sin(psi)/m11, l*cos(psi)/m33];
wek = (G3)^(-1)*(-1*F3+[-k31*dex-k32*ex; -k33*dey - k34*ey]);
tauu = wek(1);
taur = wek(2);
```

Wszystkie występujące, w implementacji sterownika, zmienne zostały opisane przy okazji dwóch wcześniejszych.

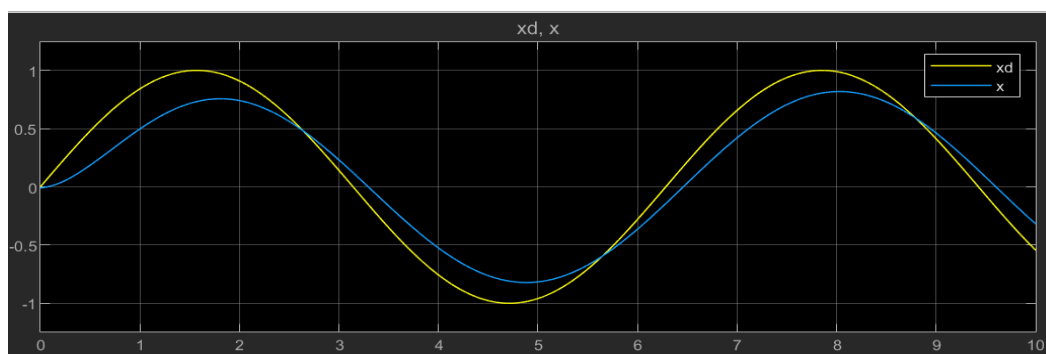
Cały schemat sterowania z wcześniej opisanymi blokami został przedstawiony poniżej:



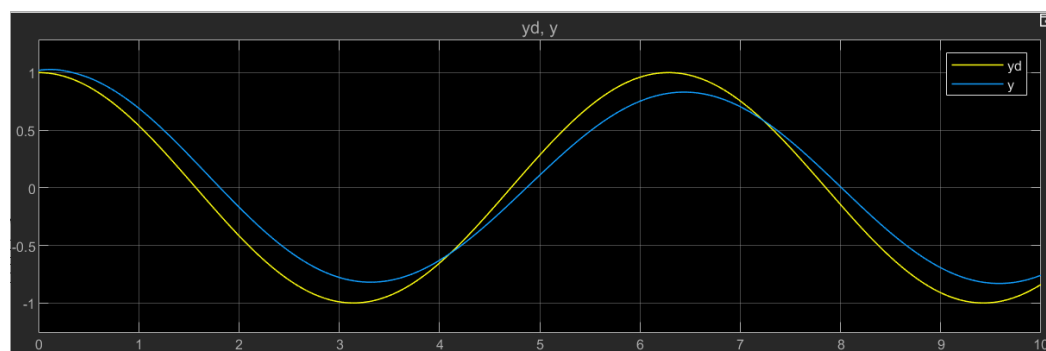
Rysunek 18 Schemat regulacji z regulatorem ORDI.

Należy zwrócić uwagę na wprowadzenie bloków przeliczających sygnał psi na wirtualne sygnały x oraz y (bloki sine, cosine oraz dwa bloki wzmacnień nad modelem statku).

Rezultaty eksperymentów:



Rysunek 19 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi X.



Rysunek 20 Trajektoria zadana(żółty) i odtworzona(niebieski) w osi Y.

6. Podsumowanie

Zbudowane układy regulacji pozwalają na odtwarzanie zadanej trajektorii w stopniu porównywalnym z artykułem[1]. W metodzie VIDI śledzenie nie działa prawidłowo na zestawie nastaw z artykułu($k_{21} = k_{22} = k_{23} = k_{24} = k_{25} = 4$), stąd wartości tych parametrów zostały zmienione na $k_{21} = k_{22} = k_{23} = k_{24} = 10$, $k_{25} = 2$.

7. Źródła

[1] Tracking control of an underactuated ship by modified dynamic inversion Linqi Ye, Qun Zong School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China