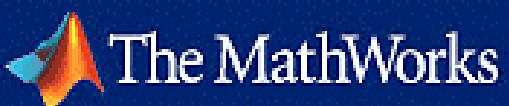




Simulink MATLAB



Przegląd obiektów i przykłady zastosowań



Simulink – MATLAB

SIMULINK jest rozszerzeniem pakietu **MATLAB**;
przy pomocy graficznego środowiska pozwala konstruować diagramy czasowe, reprezentujące określone procesy dynamiczne.



Simulink – MATLAB

W języku symulacyjnym
SIMULINK najczęściej
wykorzystuje się
pliki MATLABa (typu .m) do
rozwiązywania układów równań
różniczkowych, liniowych
i nieliniowych.



Simulink – MATLAB

Zamiast pisać kod MATLABa w SIMULINK opisujemy procesy w postaci połączonych bloków, wykorzystując odpowiednie ikony, reprezentujące dane wejściowe (input), części układu i dane wyjściowe (output).

Simulink → Simulation and Link

narzędzie (toolbox) MATLAB' a
umożliwia:



modelowanie

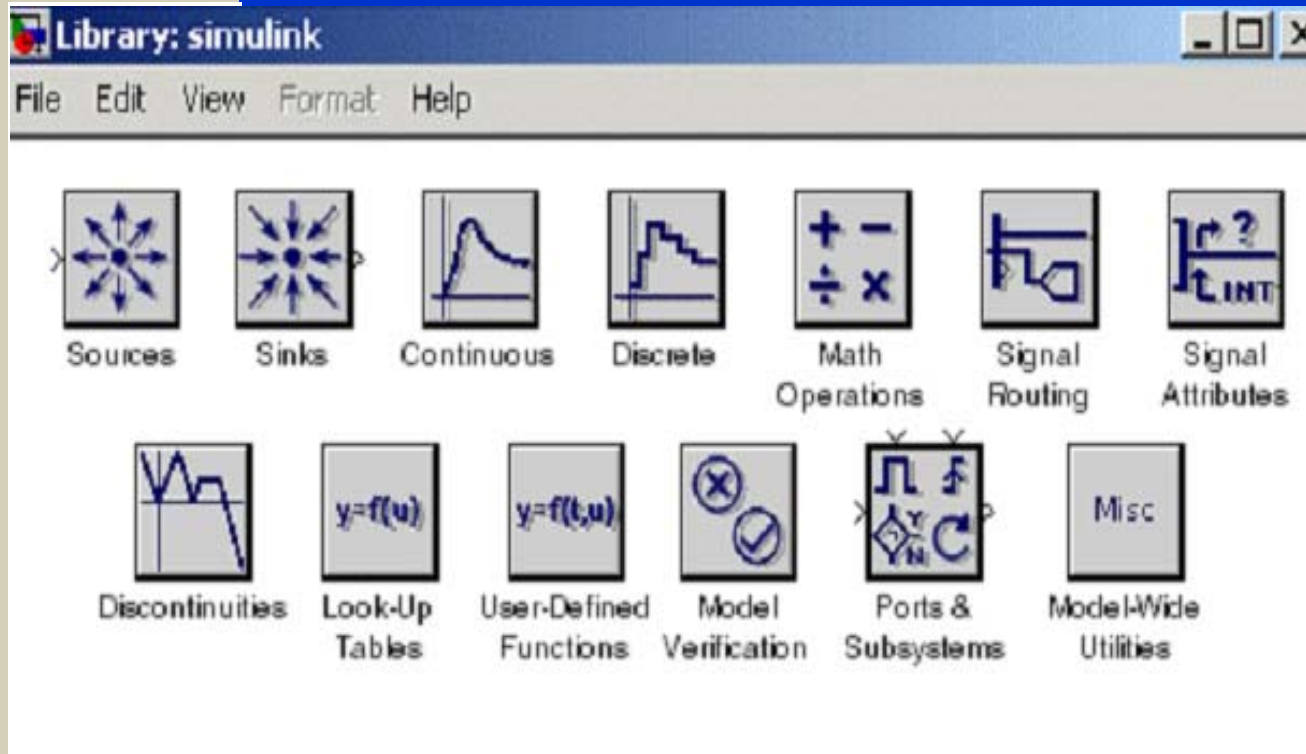


symulacje

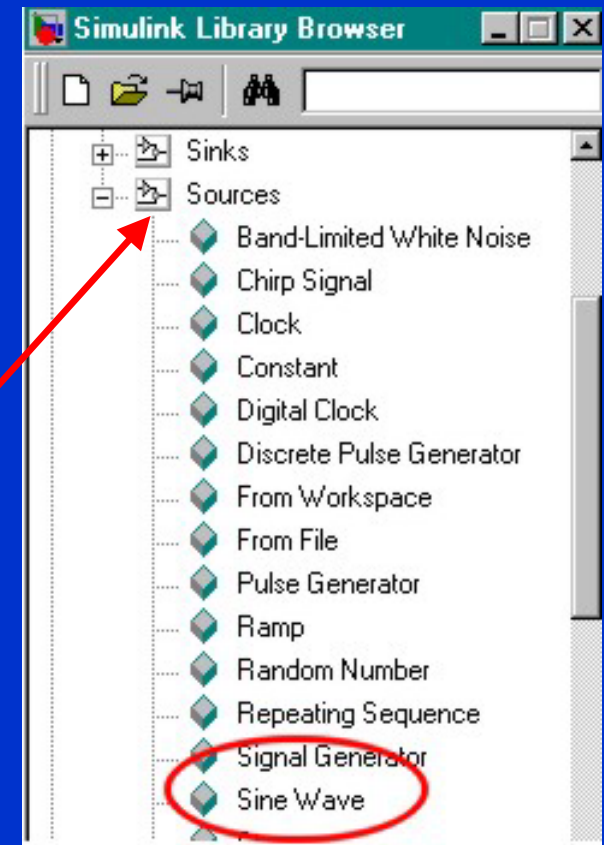
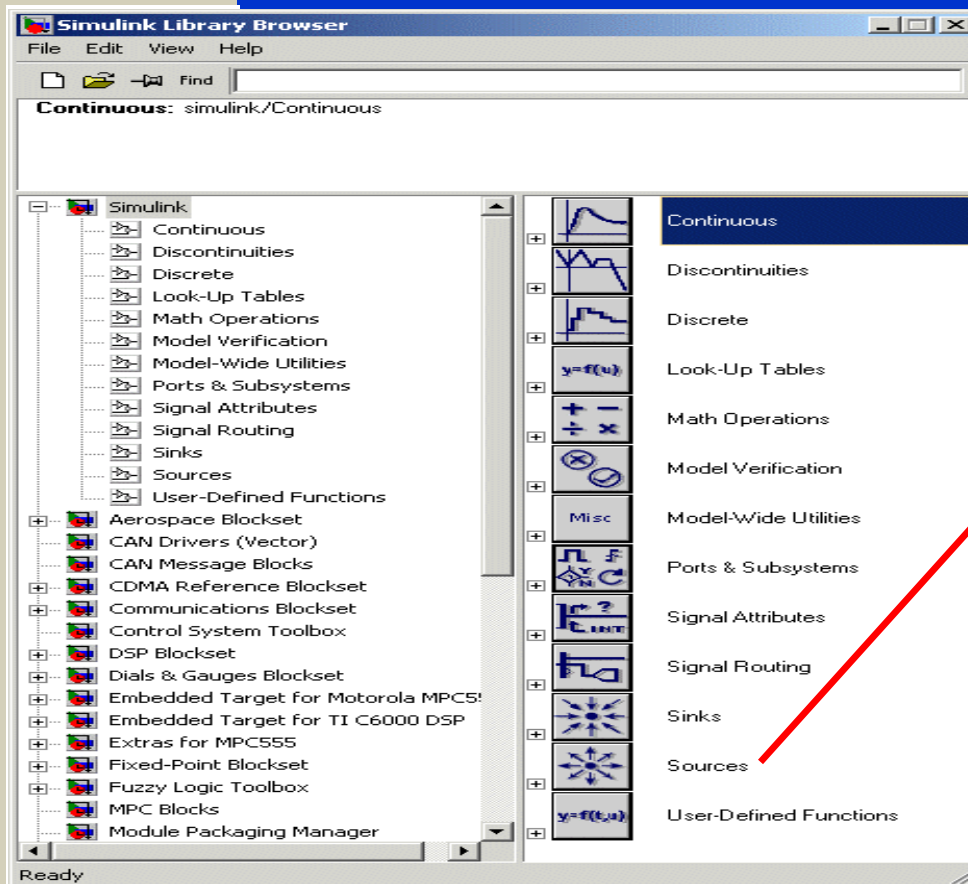


analizę układów dynamicznych

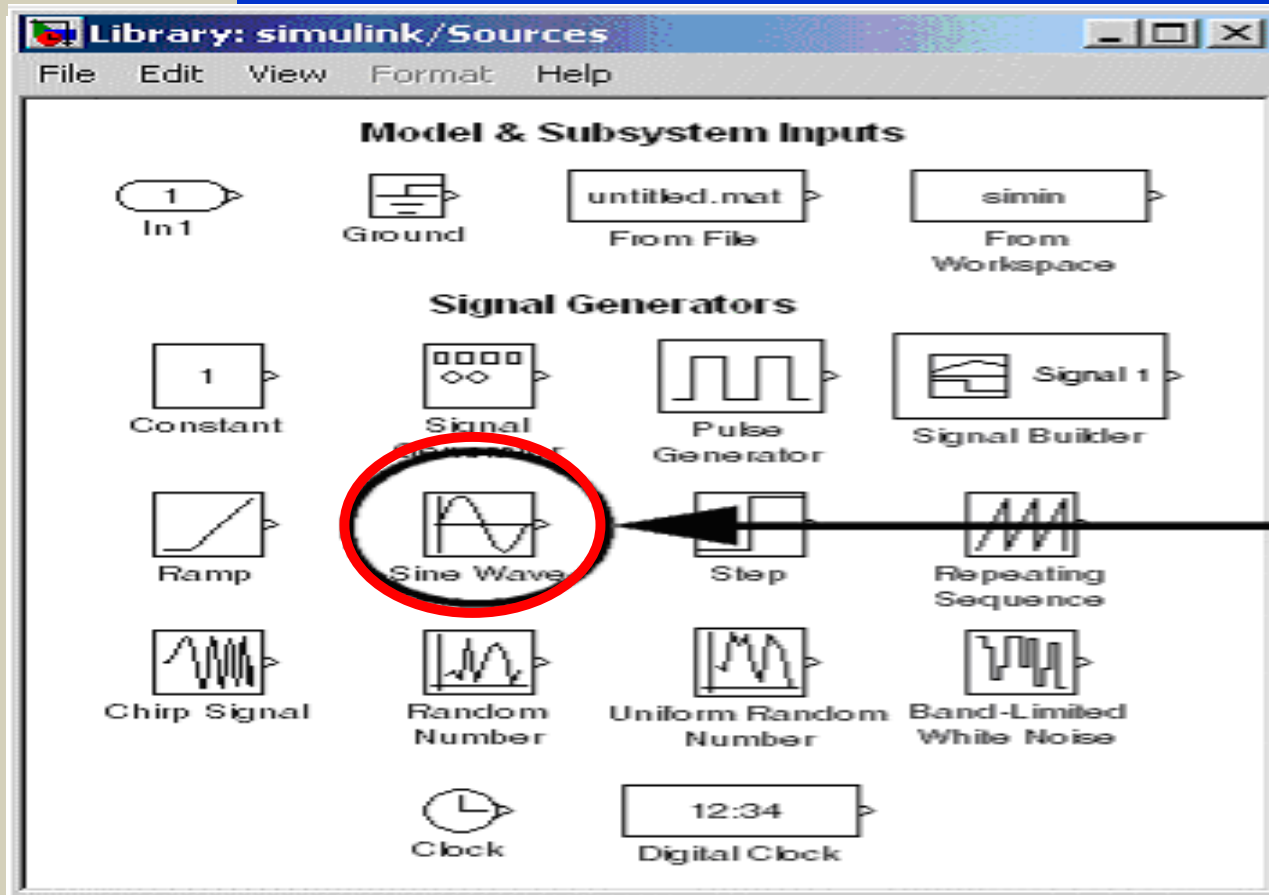
Obiekty SIMULINK - biblioteka



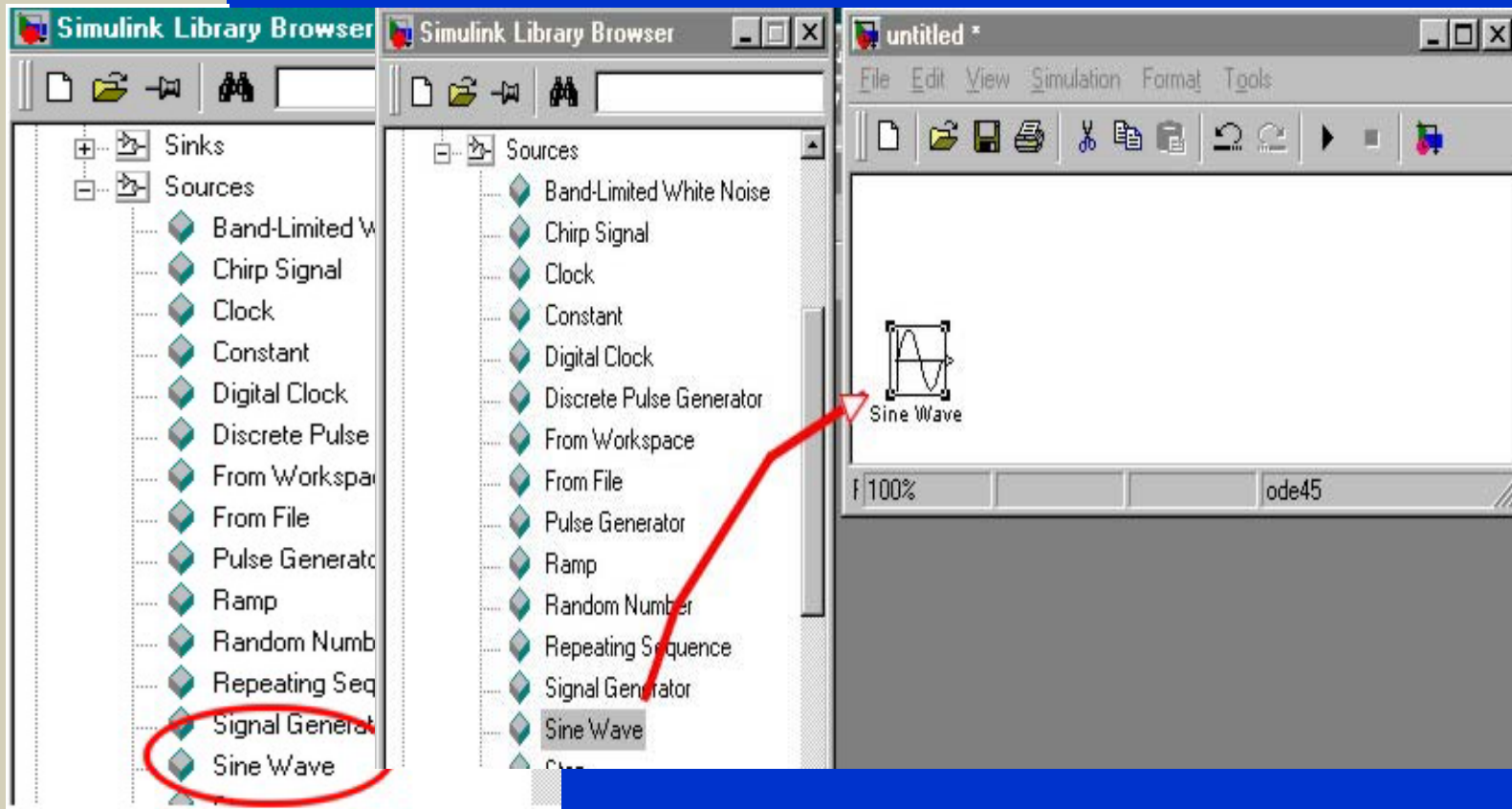
Obiekty SIMULINK - przegląd



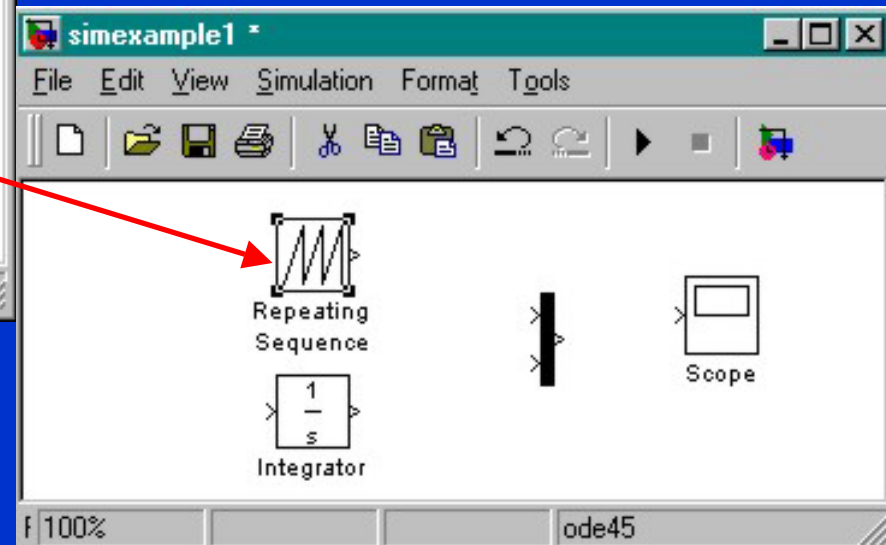
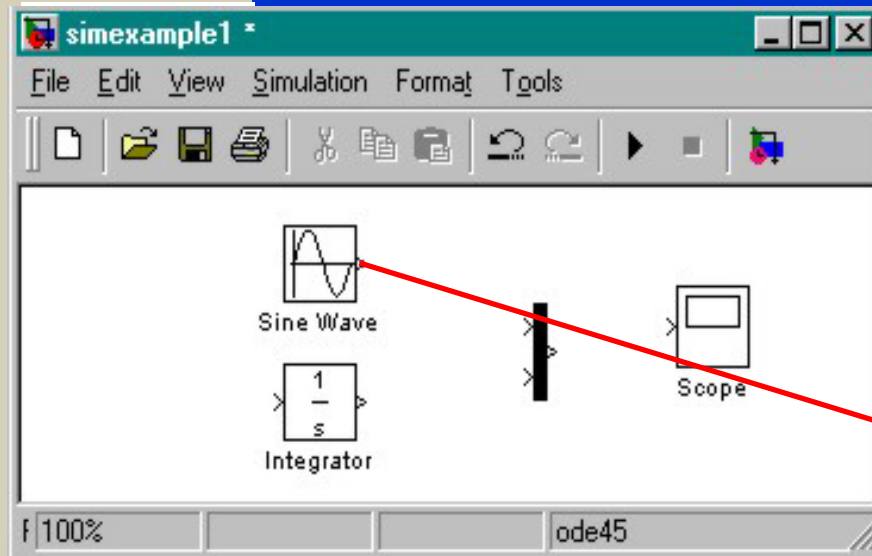
Obiekty SIMULINK - przegląd



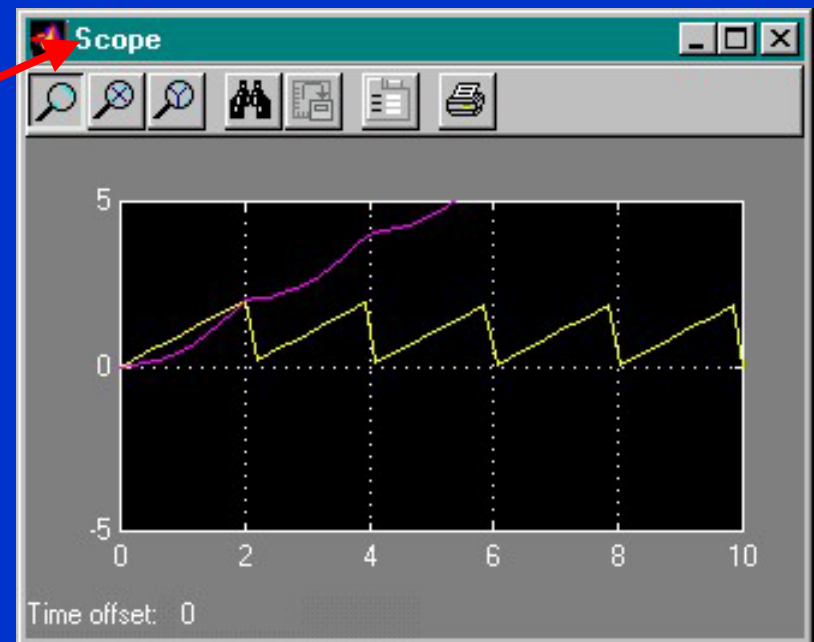
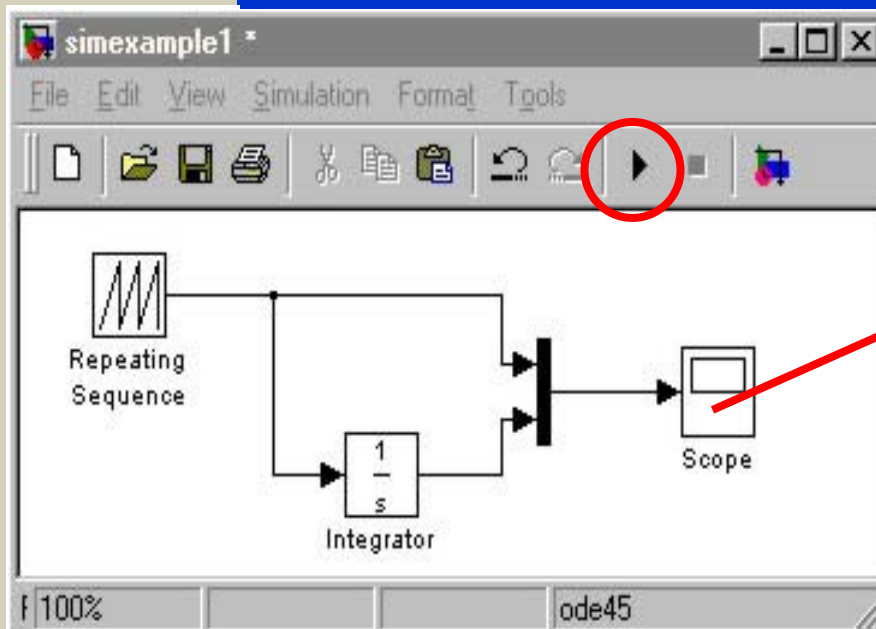
Obiekty SIMULINK - diagramy



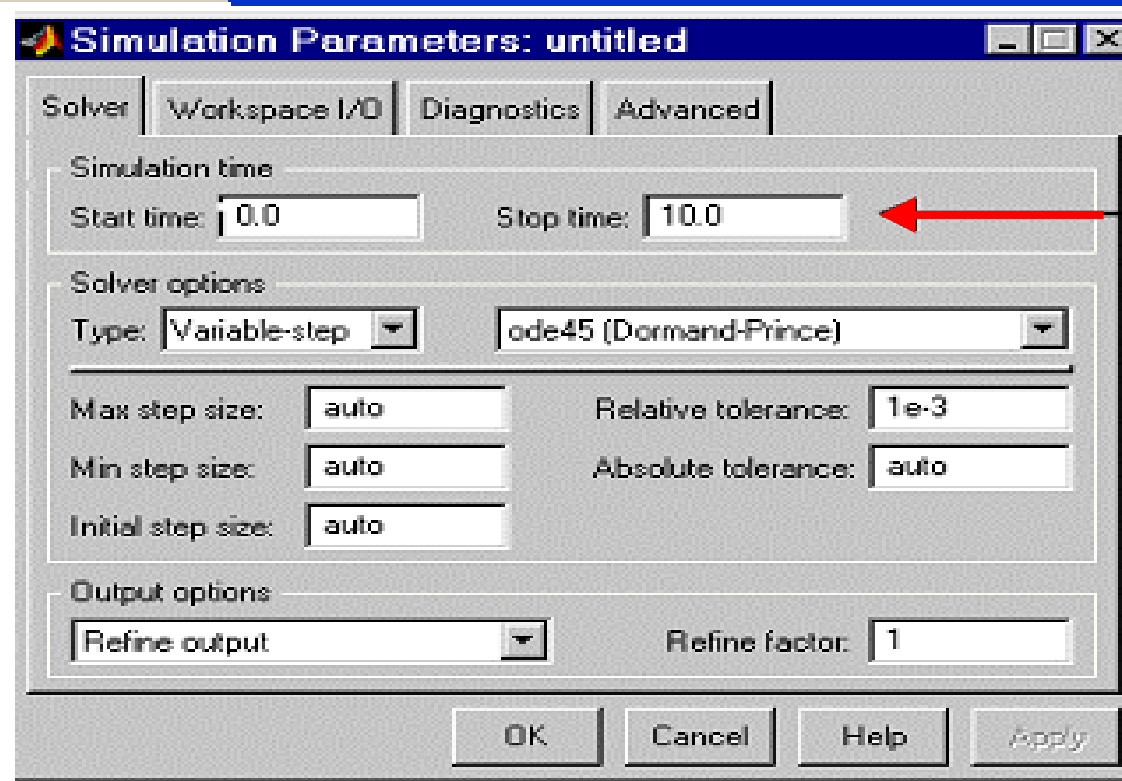
Obiekty SIMULINK - diagramy



Obiekty SIMULINK - symulacje



SIMULINK - ustawienia symulacji



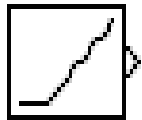
SIMULINK - ustawienia symulacji



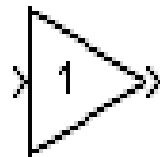
Przykłady prostych modeli

Konwersja Celsius → Fahrenheit

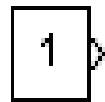
$$T_F = 9/5 (T_C) + 32$$



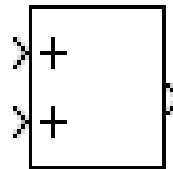
Ramp



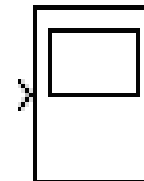
Gain



Constant



Sum



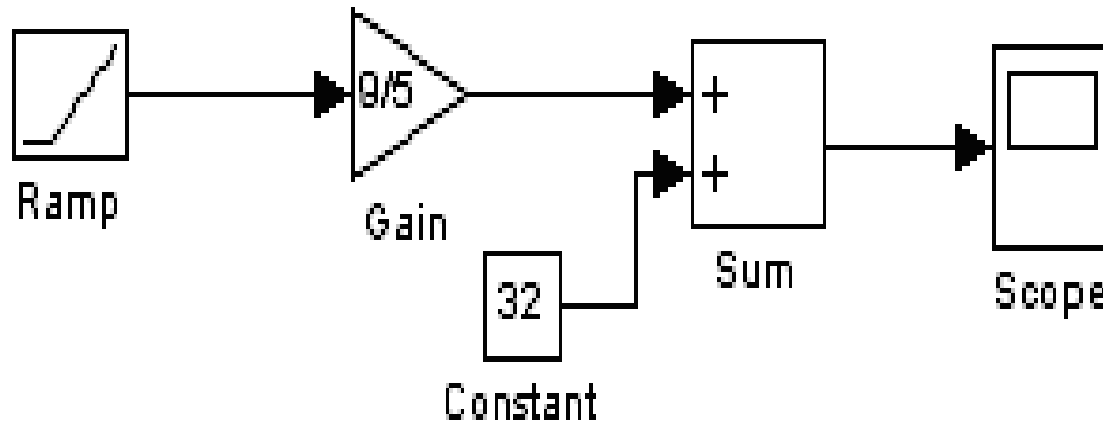
Scope

Bloki diagramu

Przykłady prostych modeli

Konwersja Celsius → Fahrenheit

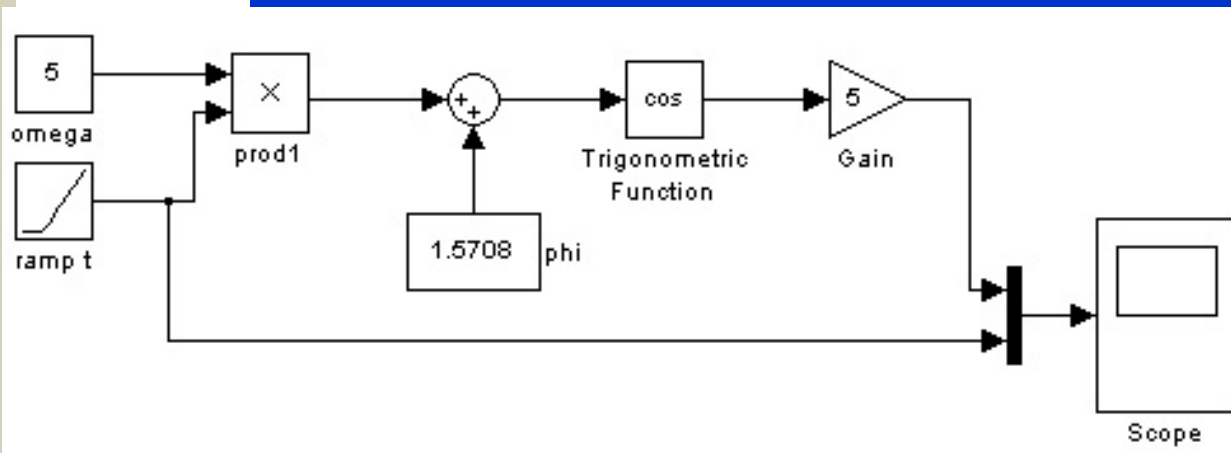
$$T_F = 9/5 (T_C) + 32$$



Połączenia bloków

Przykłady prostych modeli

$$x(t) = A \cos(\omega t + \varphi)$$



Przykładowy kod Matlab:

```
>> t=(0:.01:10);A=2;phi=pi/2;omega=5;  
>> xt=A*cos(omega*t+phi);  
>> plot(t,xt);grid
```


Prawa Newtona



równanie

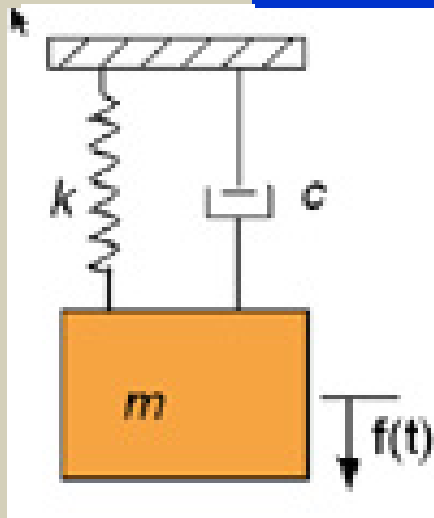


diagram modelu



wyniki symulacji

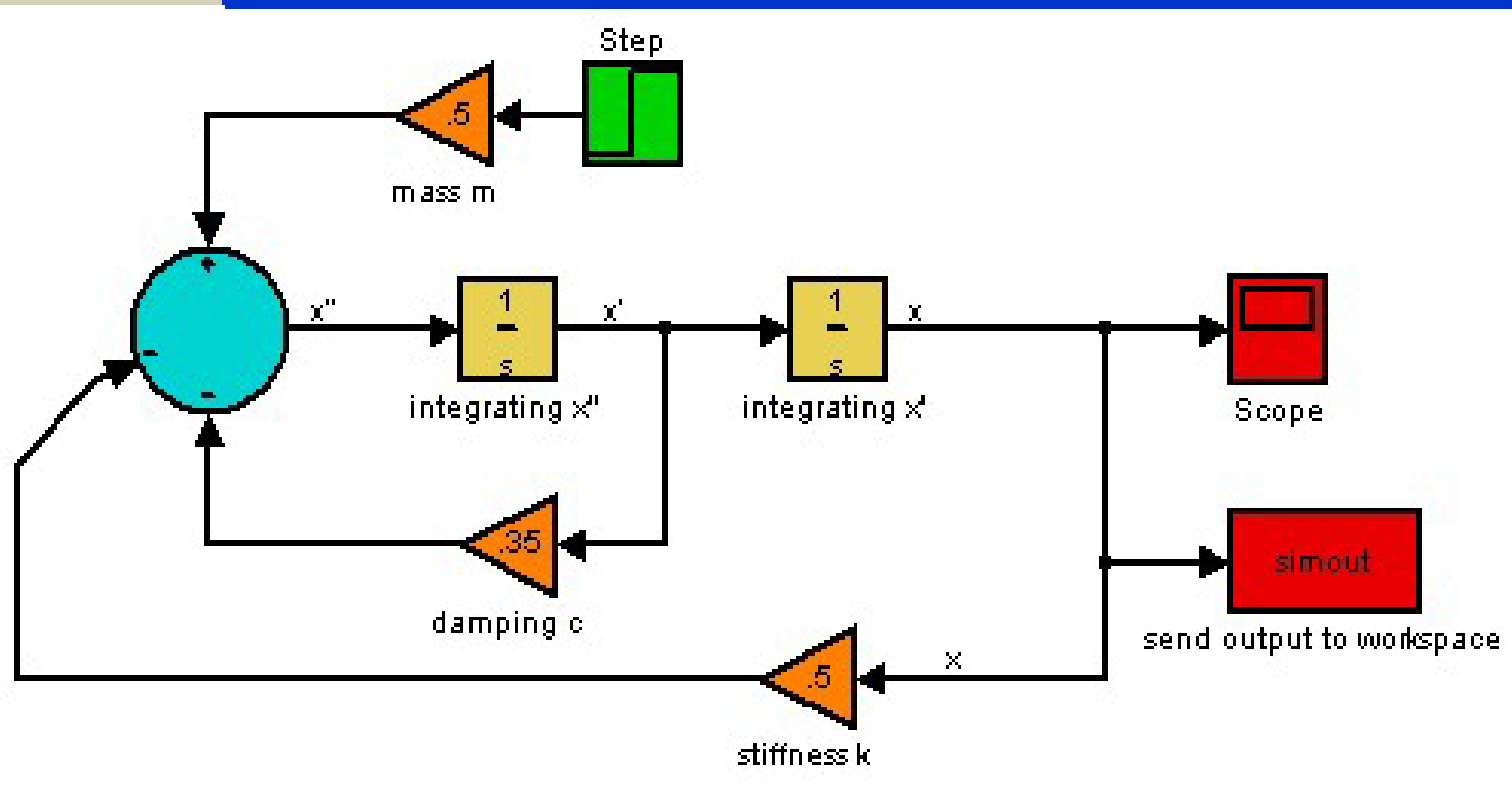
Równanie



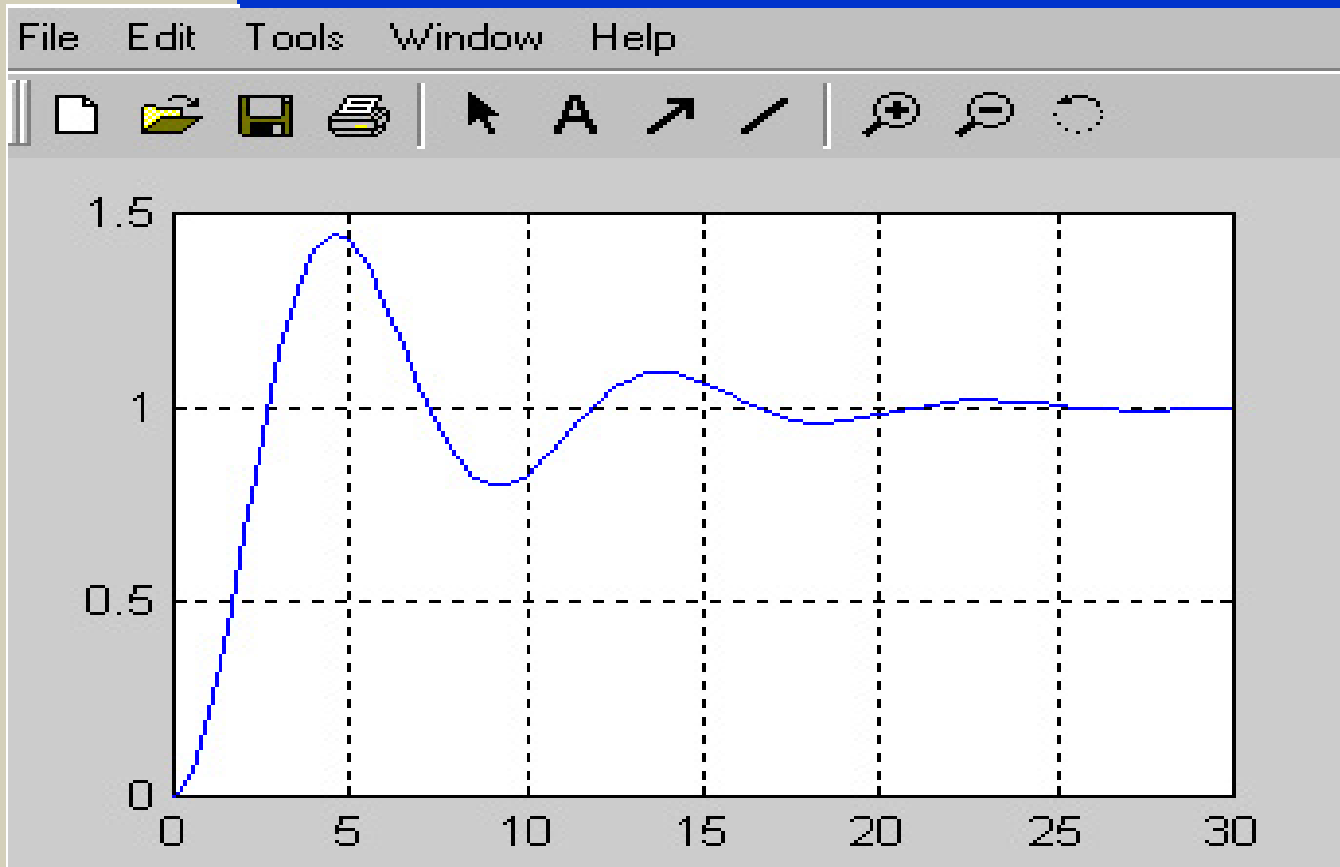
$$m\ddot{x} + c\dot{x} + kx = f(t)$$

$$\ddot{x} = \frac{1}{m} (f(t) - c\dot{x} - kx)$$

Diagram



Wyniki symulacji



Termoregulacja



ZAŁOŻENIA



PODUKŁADY MODELOWE



MODEL CAŁOŚCIOWY

Model termoregulacji - założenia

Model instalacji cieplnej domu:



termostat ustawiony na 70 stopni



Farenheita jest narażony na zewnętrzne, dzienne fluktuacje temperatury



fluktuacje temperatury modelowane przy pomocy funkcji sinusoidalnych

Model termoregulacji - moduły

W modelu zastosowano podukłady modelowe w celu:

-  uproszczenia analizy całości
-  uzyskania samodzielnych modułów do ponownego wykorzystywania

Model termoregulacji - moduły

Zastosowane podukłady:



TERMOSTAT

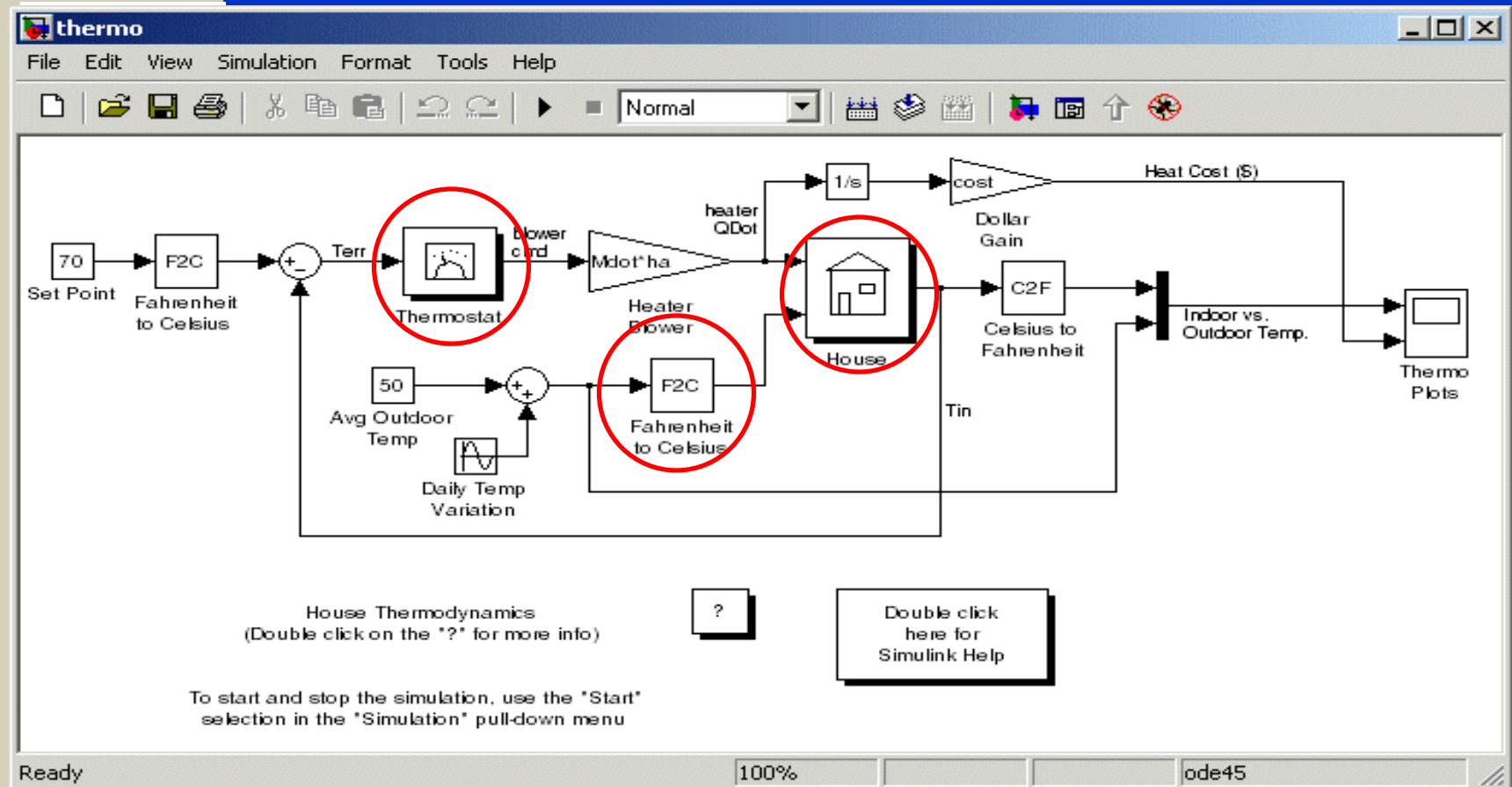


DOM

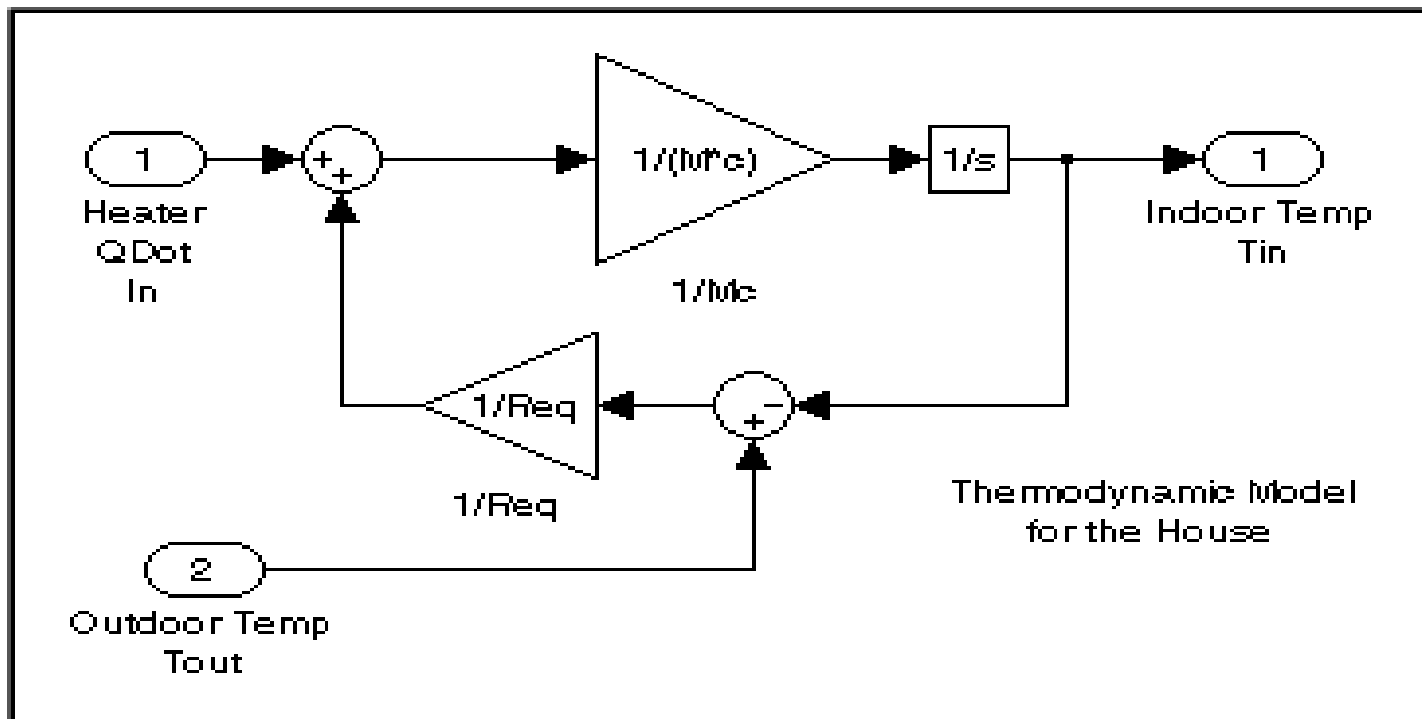


KONWERTERY TEMPERATURY

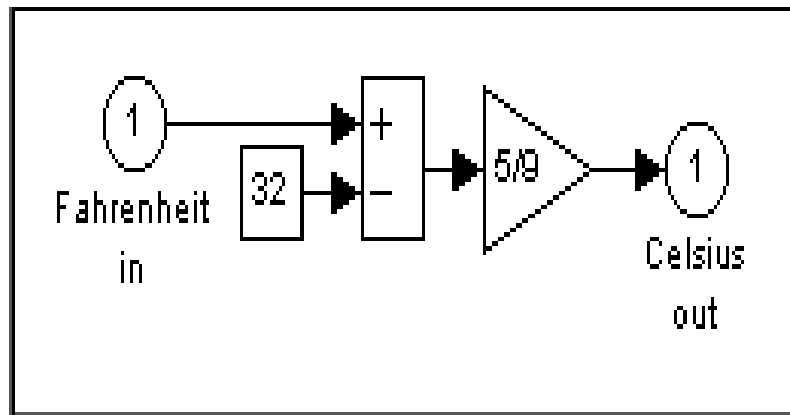
Diagram modelu termoregulaciji



Model podkładu dom



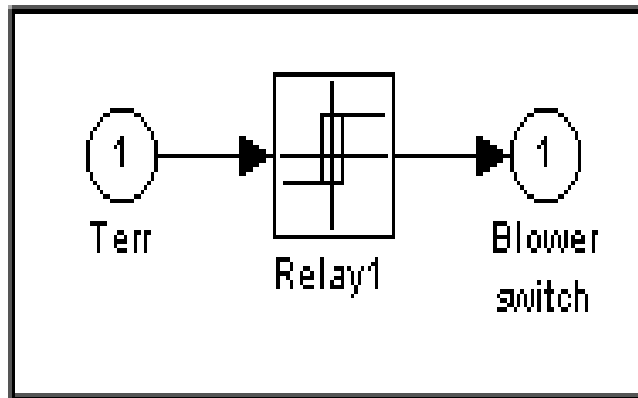
Model podukładu konwersja



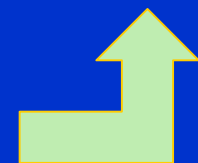
Fahrenheit to Celsius conversion (F2C)



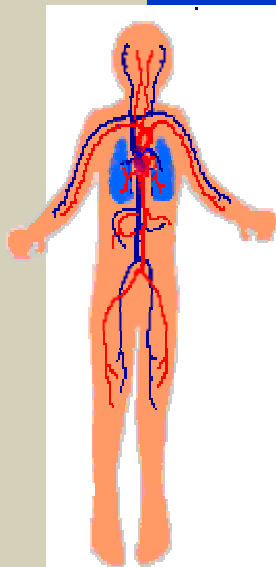
Model podukładu termostat



Thermostat subsystem

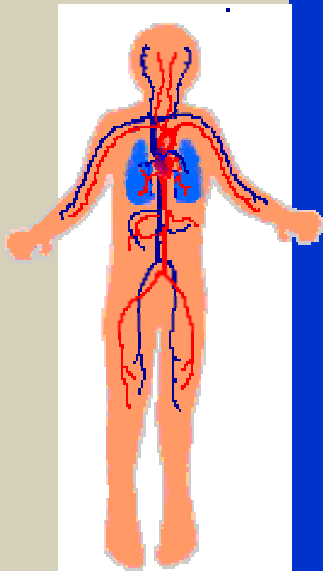


PHYSBE – w Simulink MATLAB



zaawansowana symulacja układu krążenia

Physbe



Idea



Wersja Simulink MATLAB

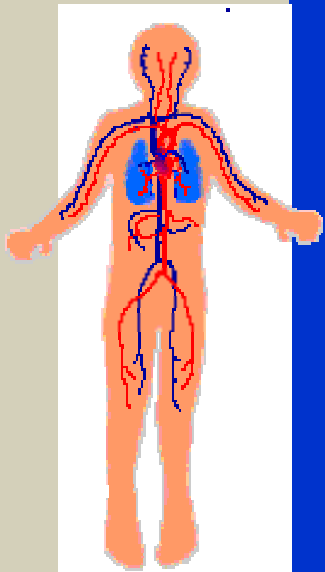


Moduły



Rezultaty

Physbe - idea



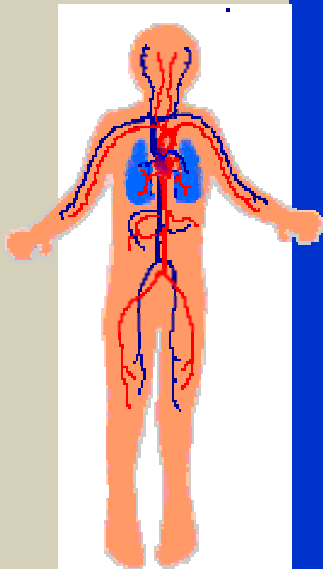
Klasyczny model układu krążenia człowieka

John McLeod 1966

PHYSBE - .. a PHYSiological simulation
BEnchmark experiment

[SIMULATION, 7, 324-329 (1966)]

Physbe - idea

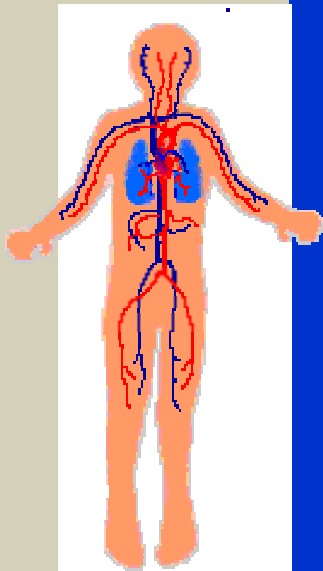


Przepływ krwi

– zgodny z mechanizmem transportu -
generuje przepływy:

- ☀ ciepła
- ☀ tlenu
- ☀ składników odżywczych

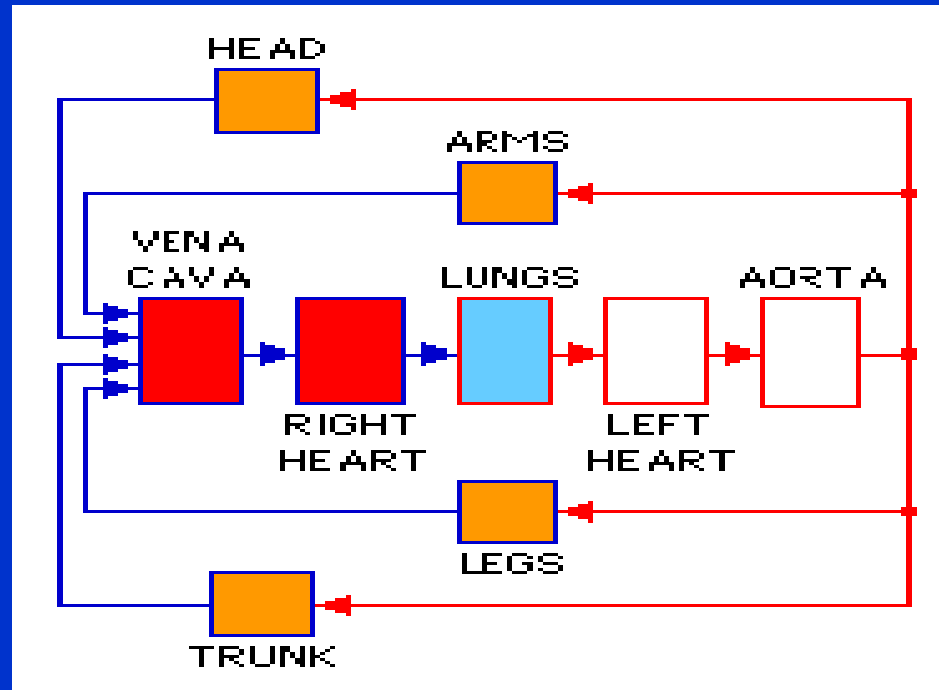
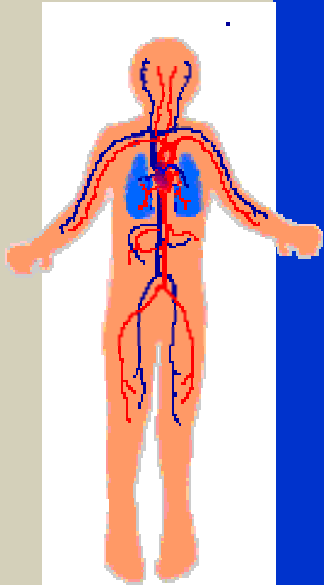
Physbe – idea



**Próby implementacji modelu
podejmowane przez wielu autorów
w latach 1966-1996**

**Wersja Simulink MATLAB
dostępna za darmo
do adaptacji własnej !!**

Physbe – schemat klasyczny



Physbe – ograniczenia modelu



niewzględnianie gęstości krwi



uproszczony liniowy model przepływu krwi



natychmiastowe zamykanie się zastawek

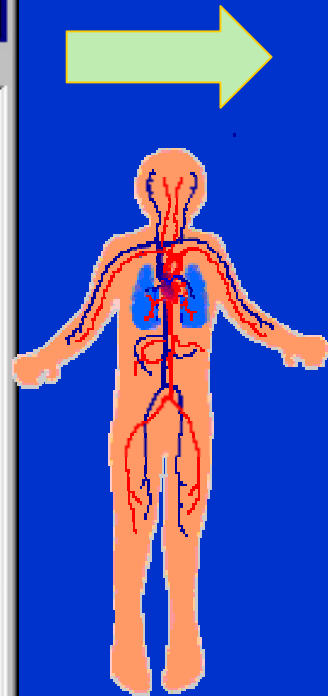
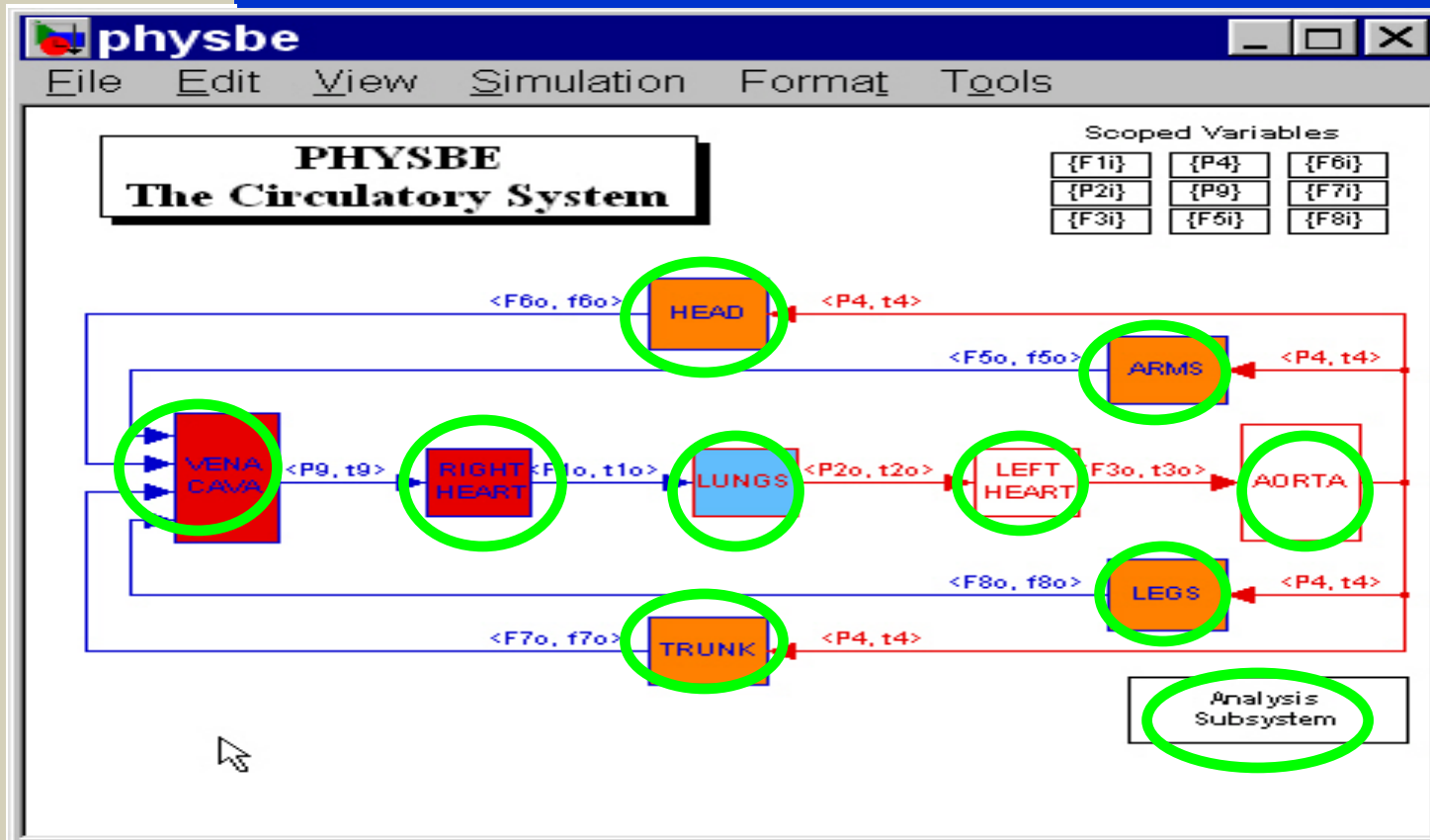


liniowe reakcje organów, naczyń krwionośnych

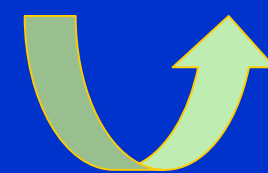
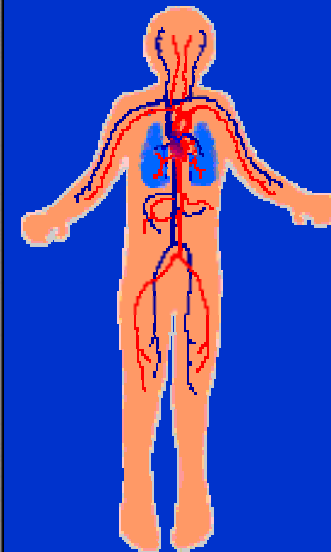
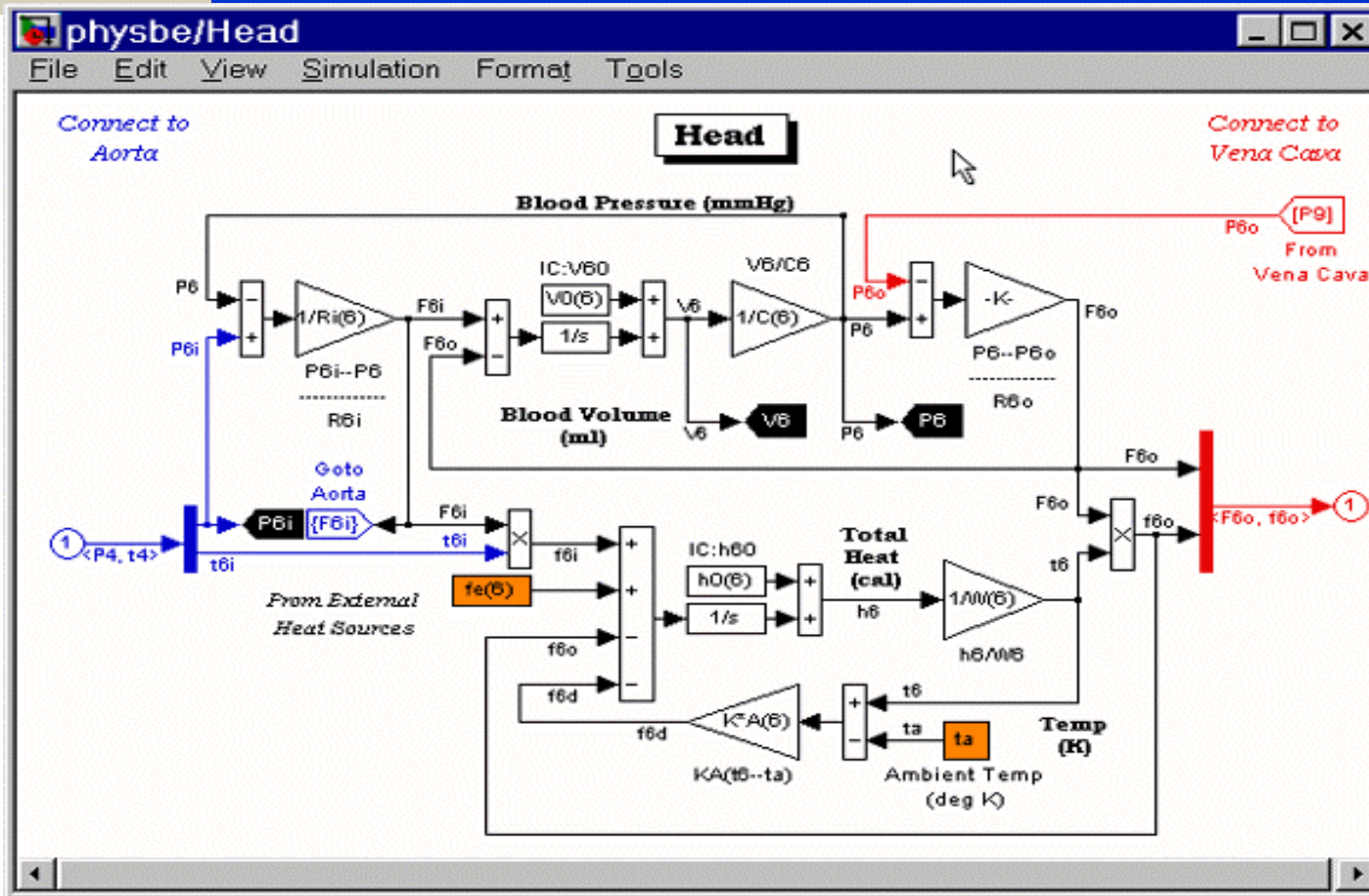


szttywny podział elementów układu

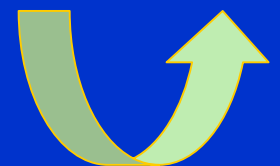
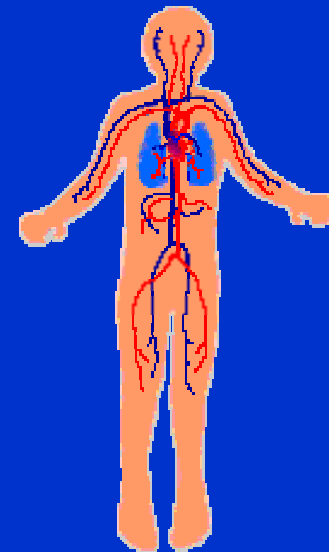
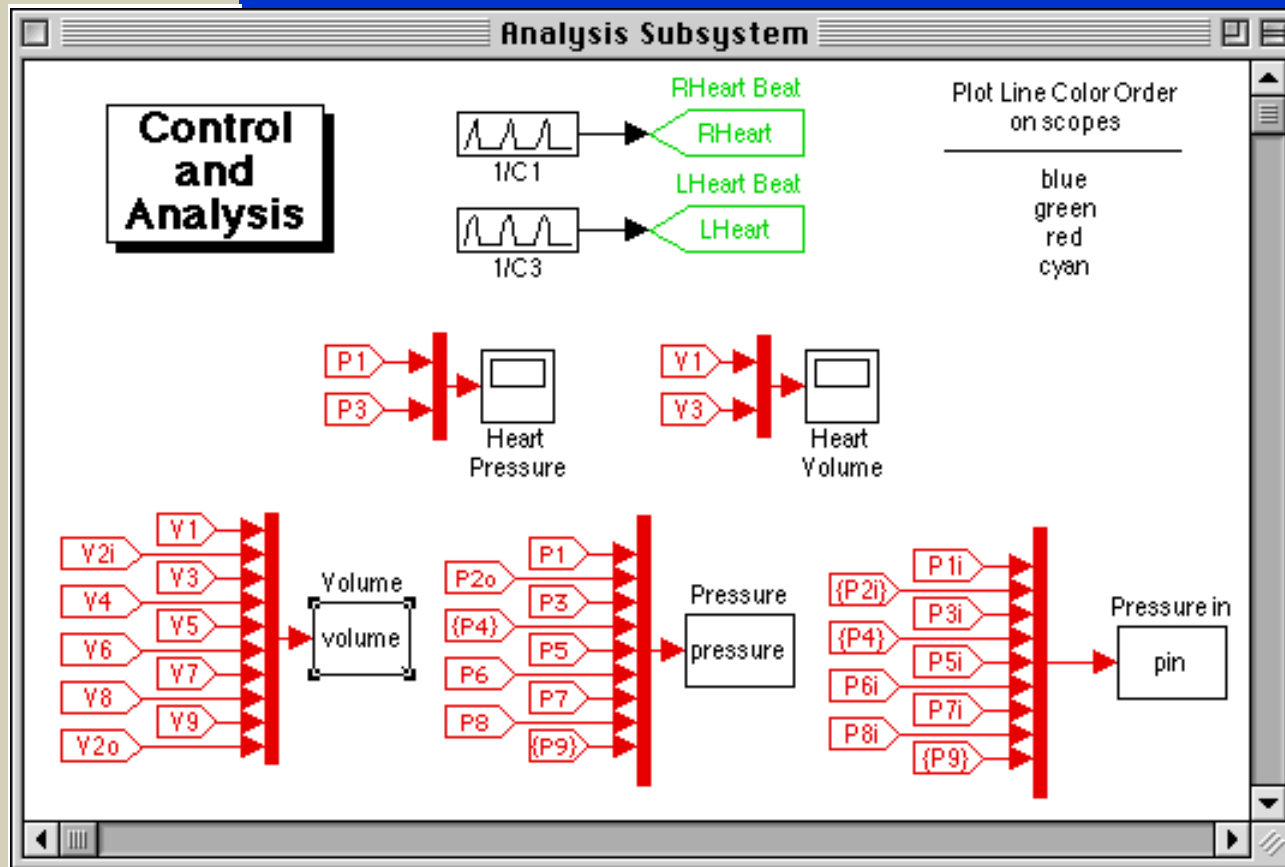
Physbe – układ krążenia SIMULINK



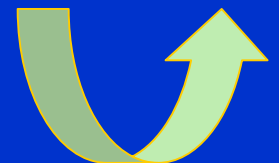
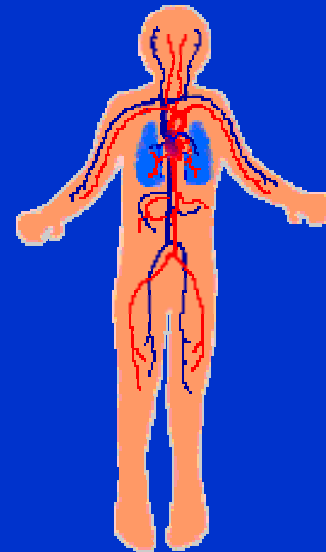
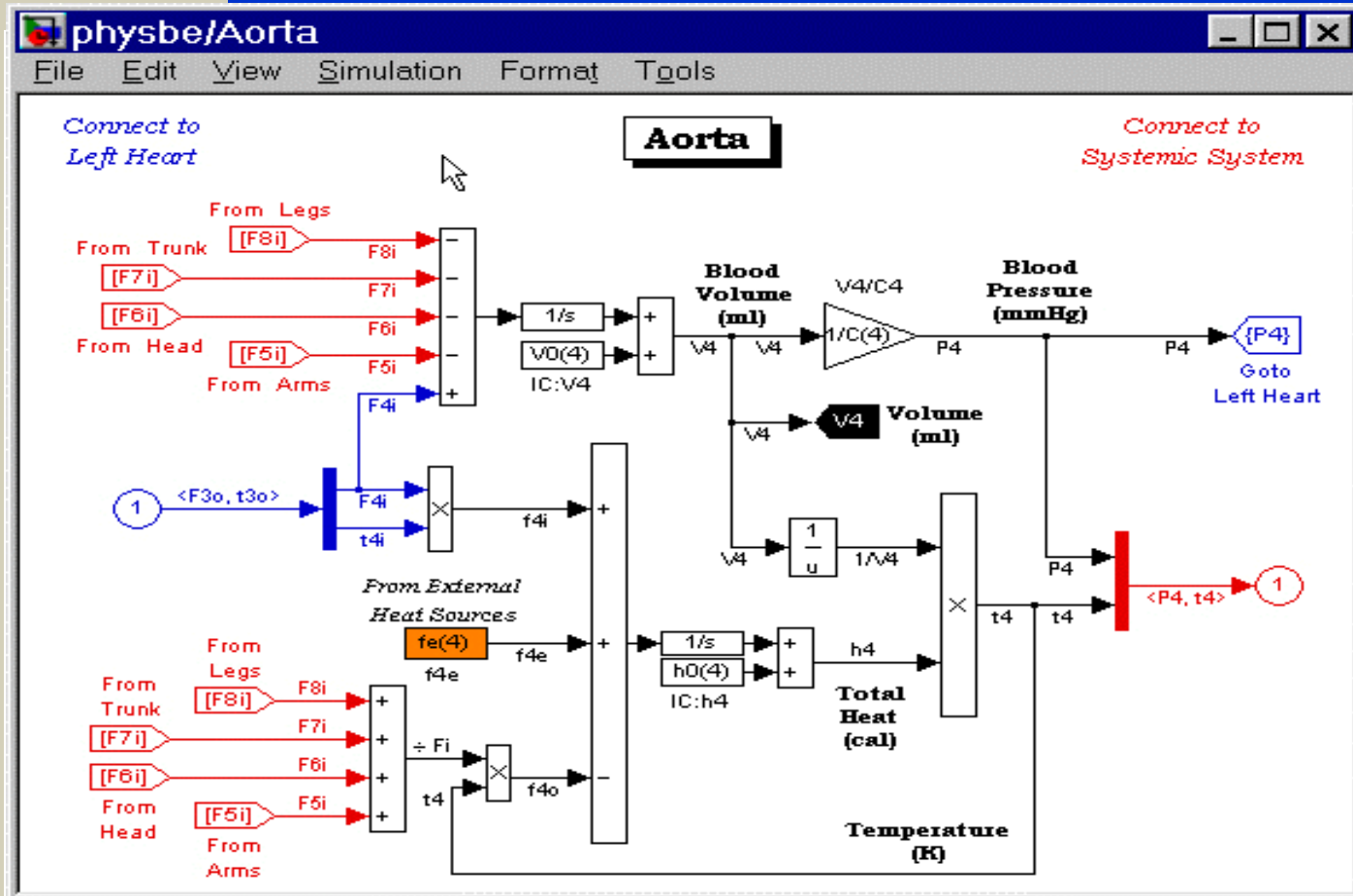
Physbe - głowa



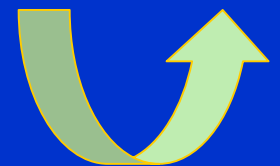
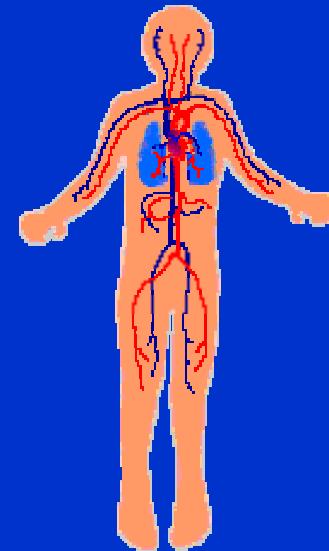
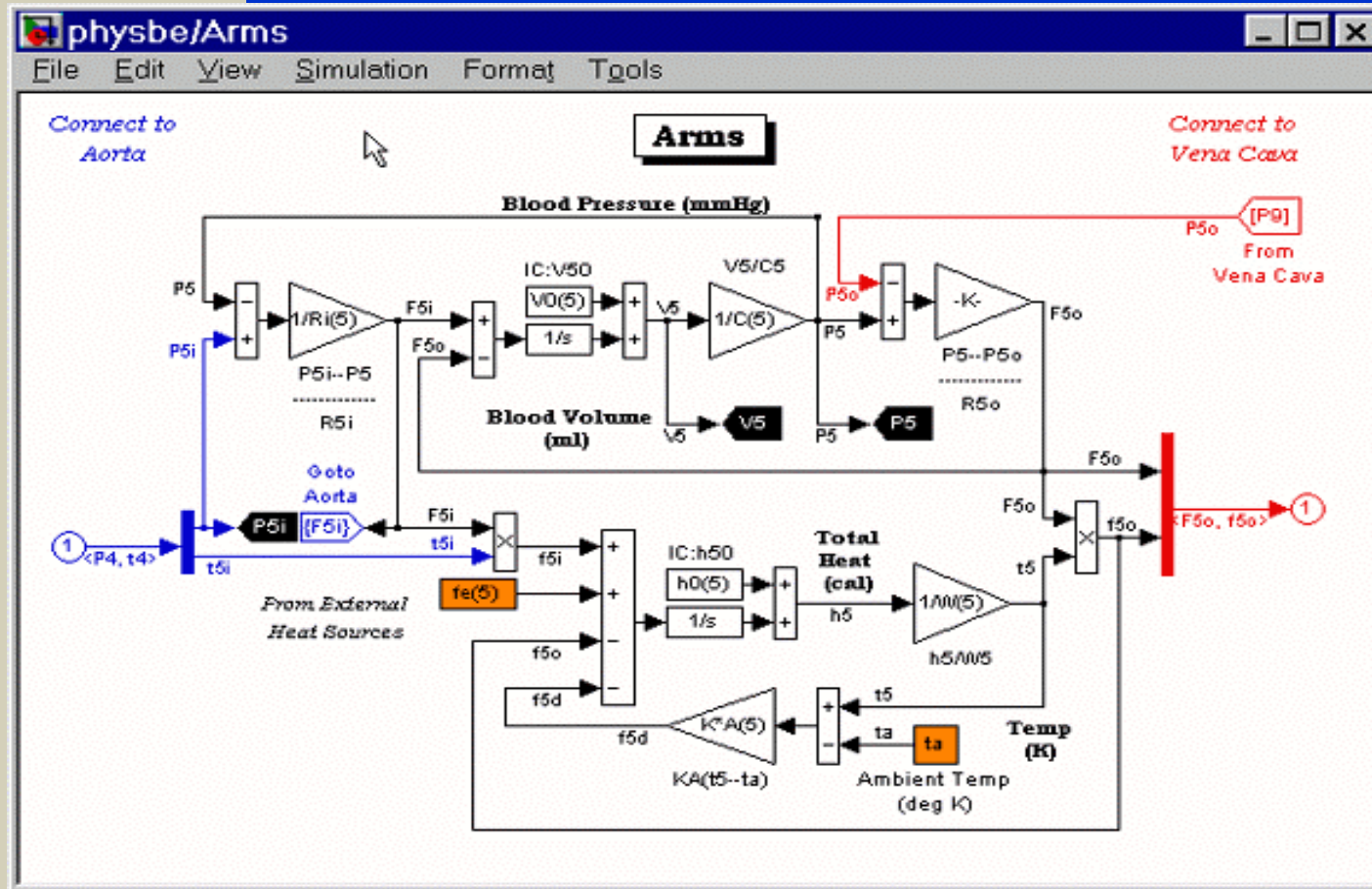
Physbe – analiza podukładów



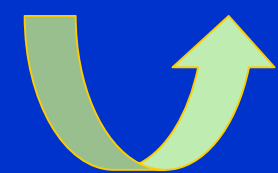
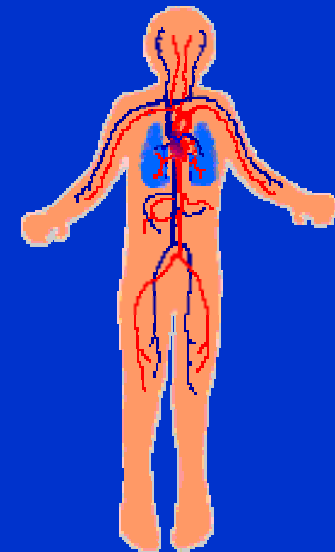
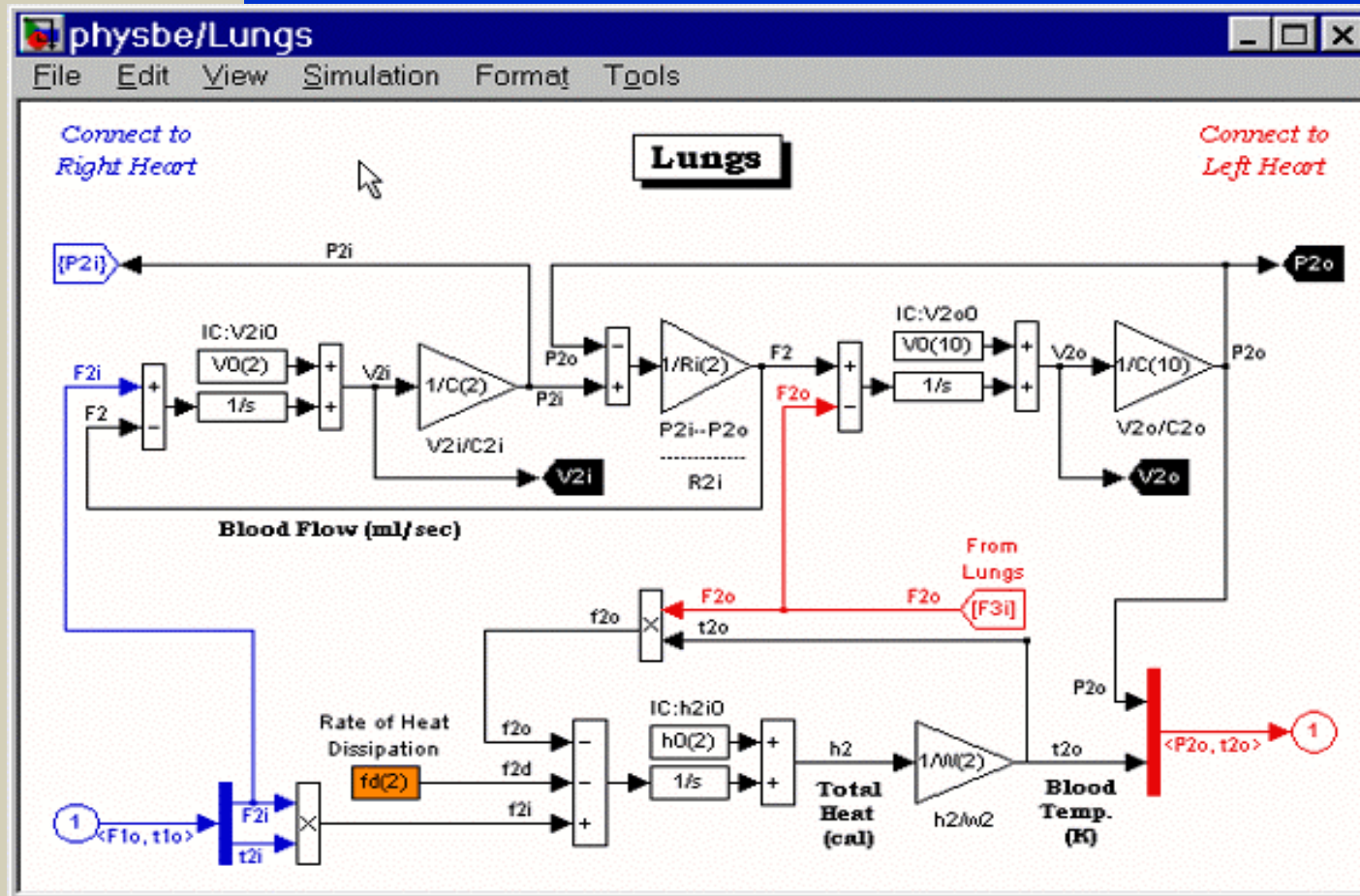
Physbe - aorta



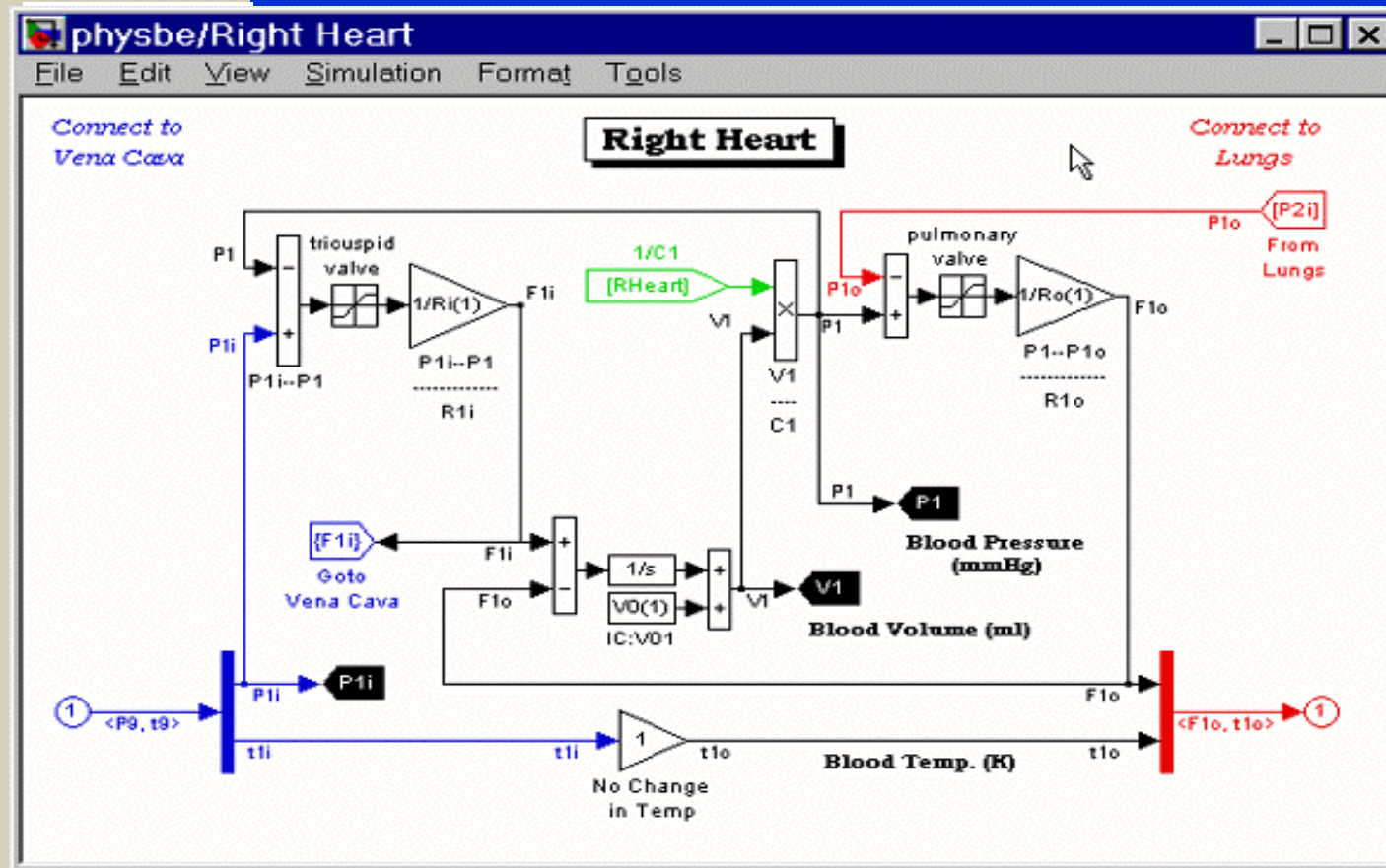
Physbe – kończyny górne



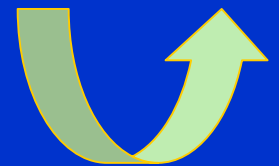
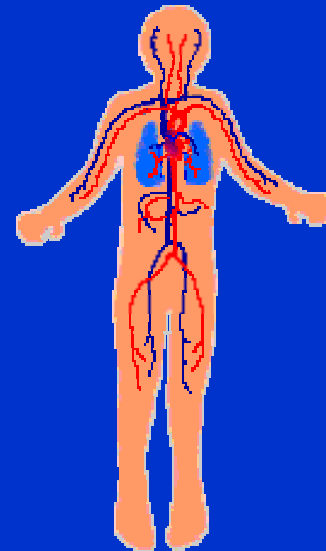
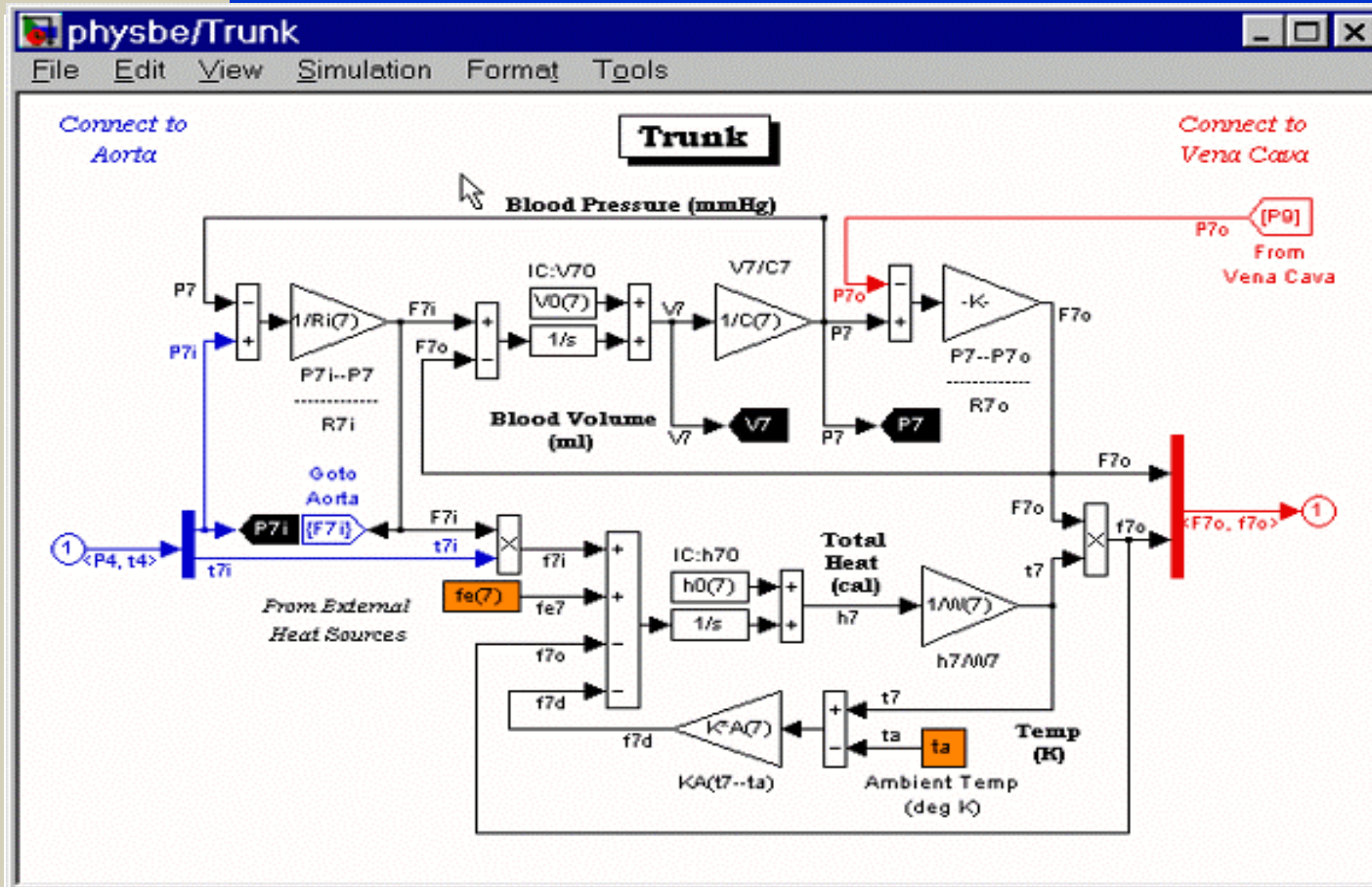
Physbe - płuca



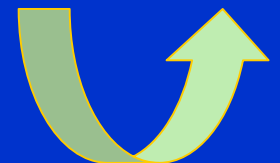
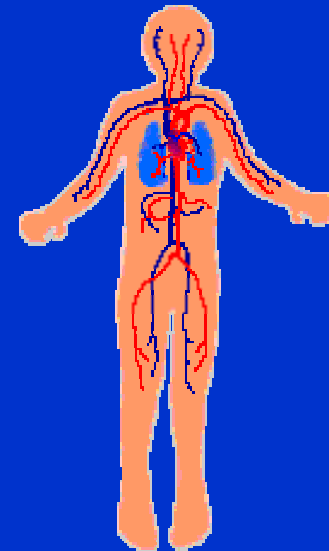
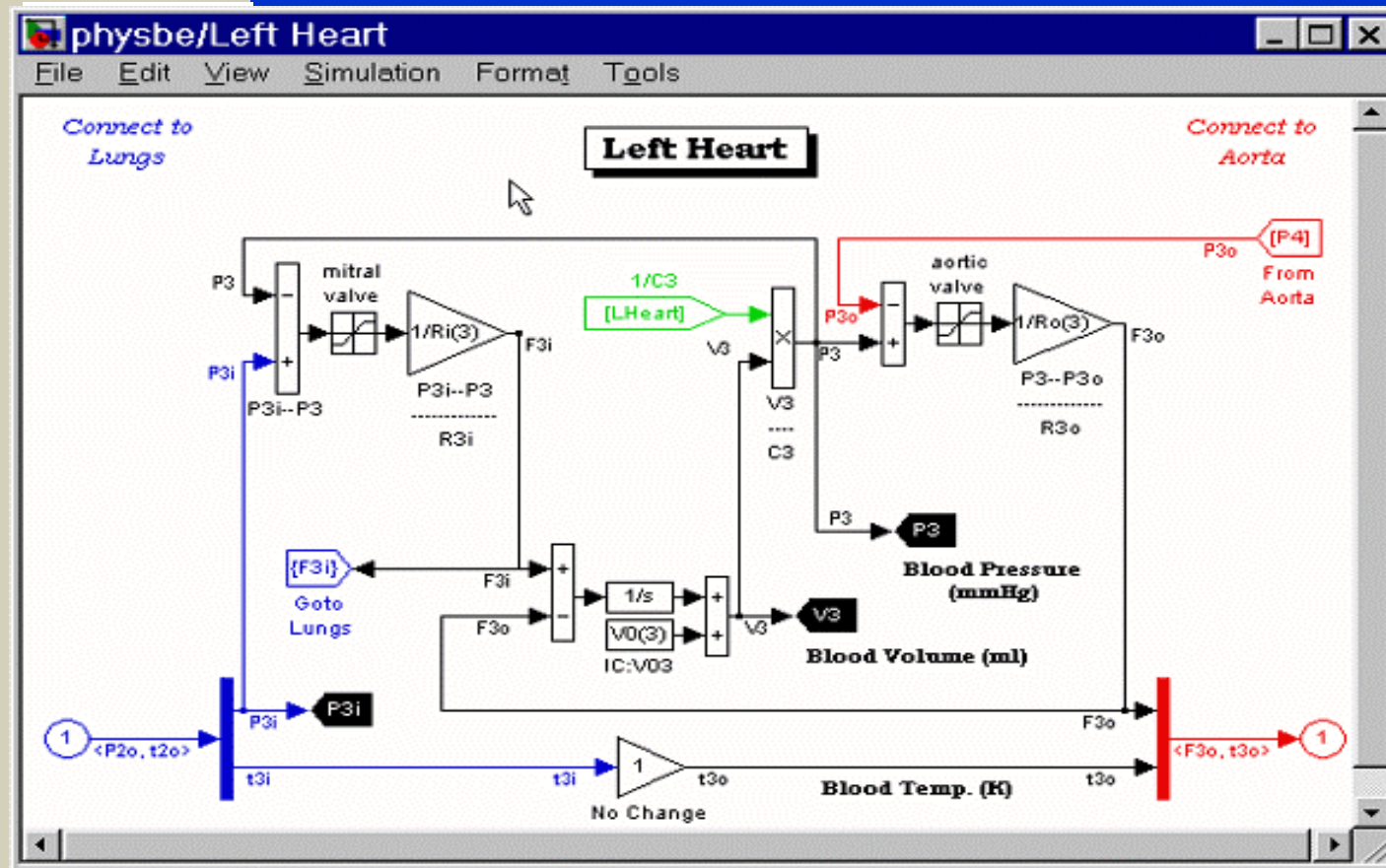
Physbe – prawa komora serca



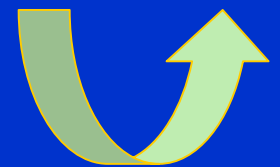
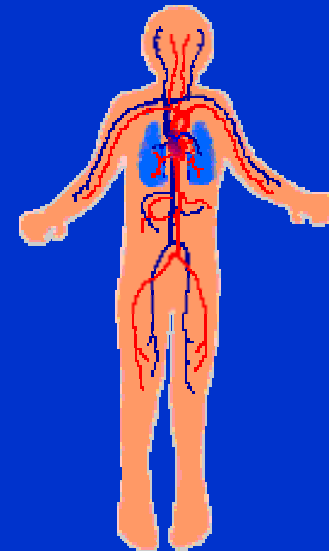
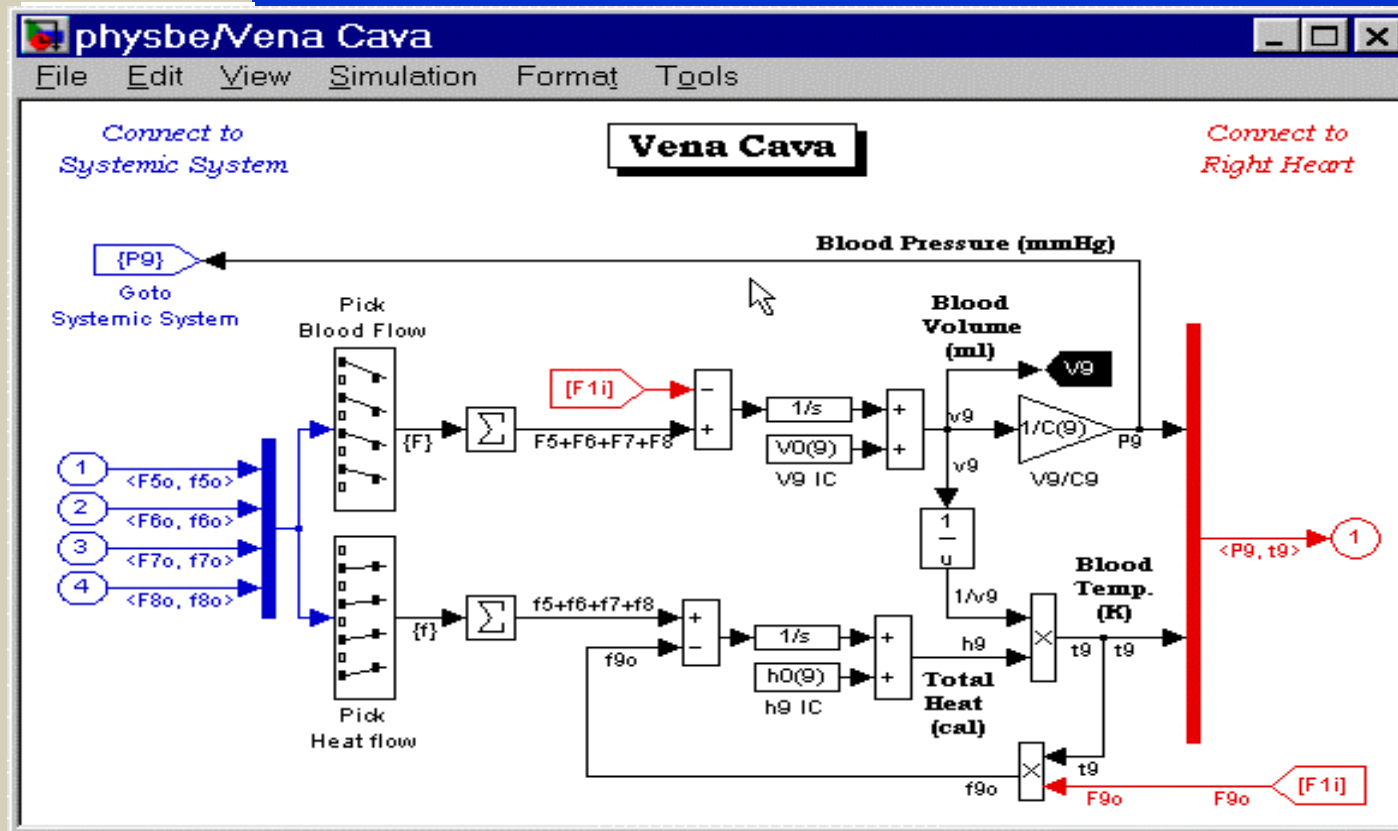
Physbe - tułów



Physbe - lewa komora serca



Physbe – żyła główna



Physbe – centrum monitorowania

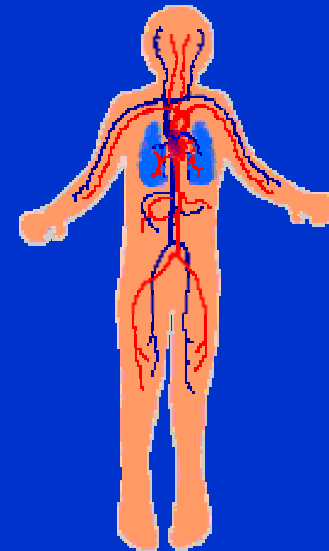
PHYSBE Control Center

File Edit Tools Window Help

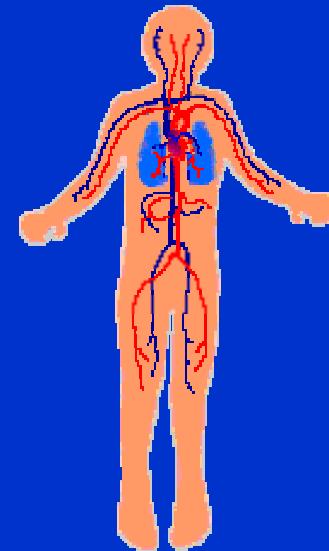
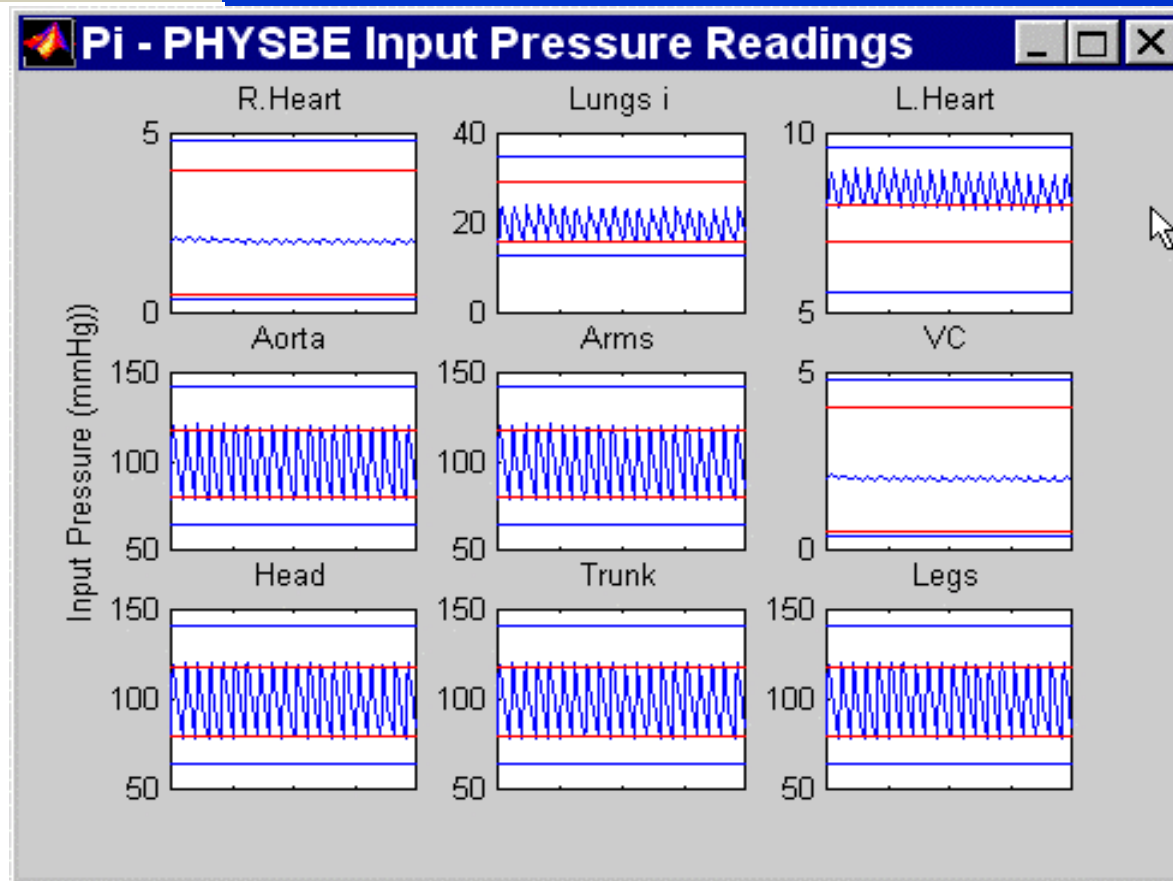
Save Reset

PHYSBE Control Center

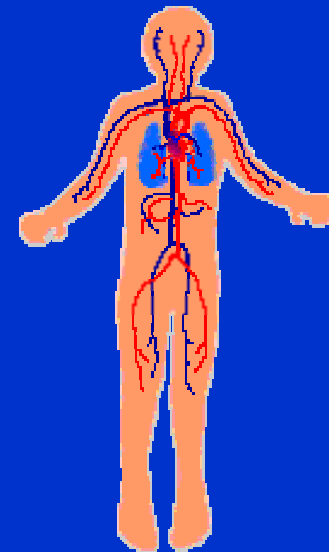
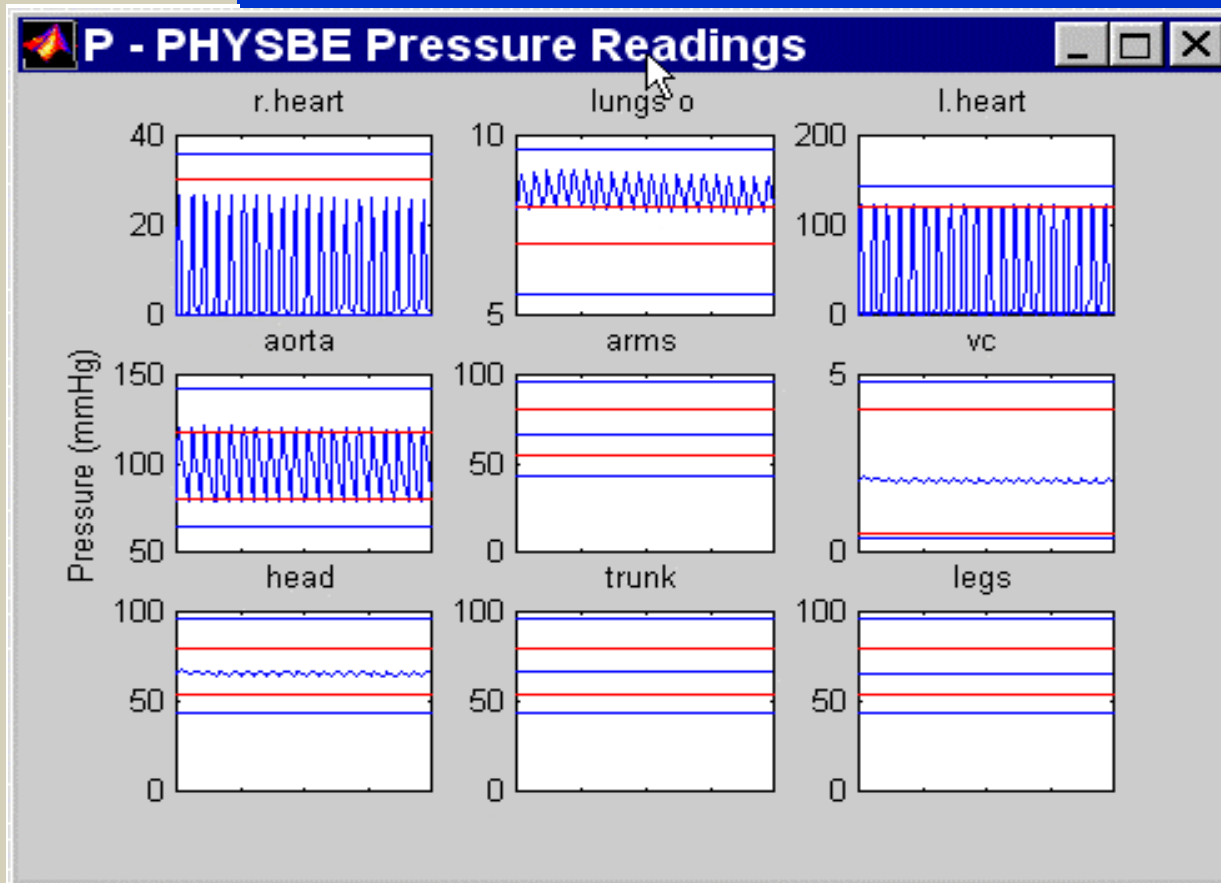
	Ri	Ro	C	W	A	V0	h0	fe	fd
R.Heart	0.0128	0.0111	75	600	0	150	0	0	0
Lung	0.1429	0	7.519	1000	0	120	0	0	0
L.Heart	0.0588	0.0125	80	600	0	150	0	0	0
Aorta	0	0	1.25	0	0	100	0	0	0
Arms	5.15	10	4.25	7000	3670	280	0	0	0
Head	2.58	5	1.21	4500	1400	80	0	0	0
Trunk	0.67	1.42	34	53000	6000	2250	0	0	0
Legs	2.58	5	11.1	18500	7000	730	0	0	0
V.Cava	0	0	250	0	0	500	0	0	0
Lung2*	0	0	30.3	0	0	240	0	0	0



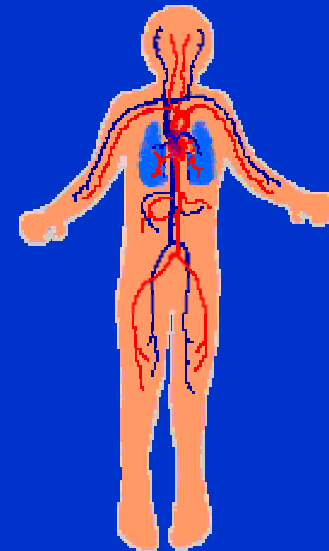
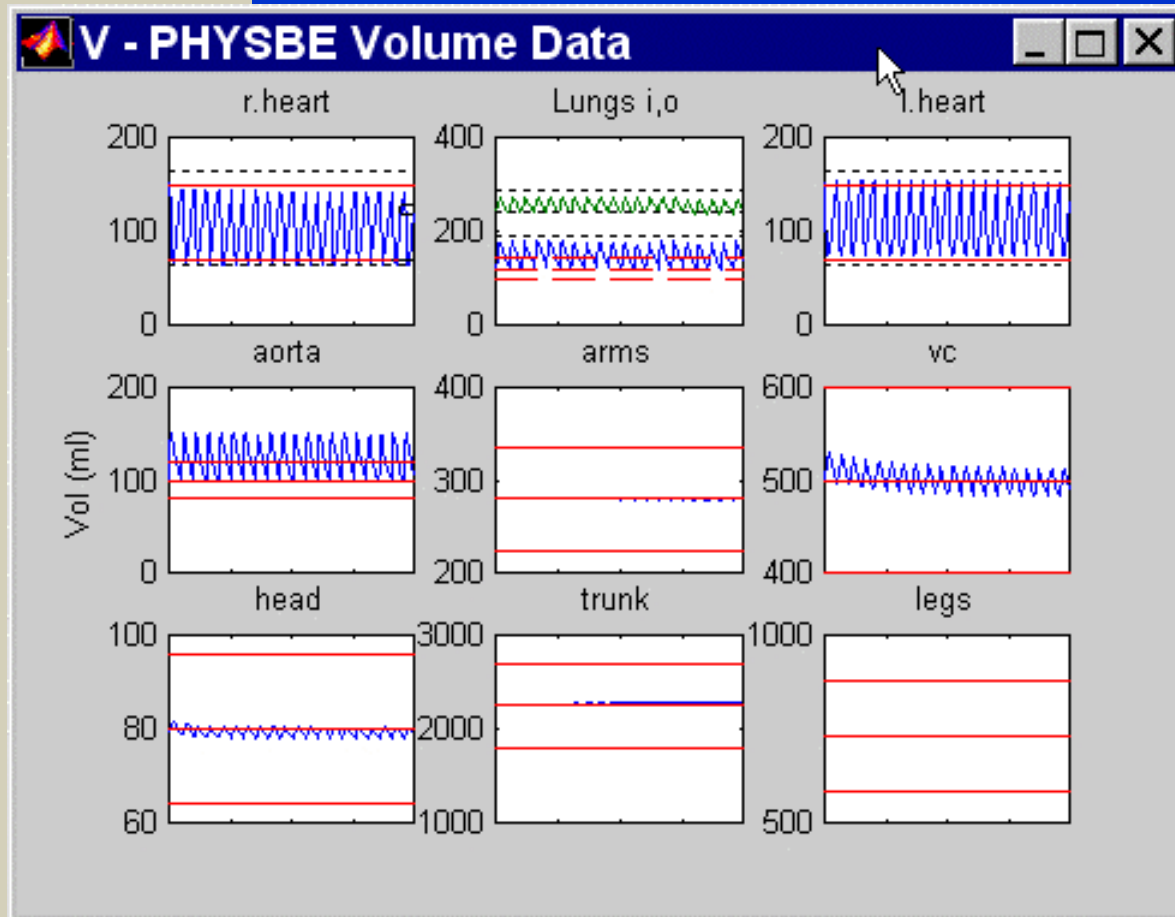
Physbe – ciśnienie - dane startowe



Physbe – kontrola ciśnienia



Physbe – kontrola pojemności



Lądowanie na księżycu w Simulink

odtworzenie historycznego
oprogramowania dla modułu lądującego



Lądowanie na księżycu



- ☀️ Oryginalny projekt modułu lądującego LM (*Lunar Module*) 1961-1963
- ☀️ Zasadnicze problemy 1963-66
- ☀️ Wersja Simulink MATLAB

Moduł lądowania na księżycu



- NASA - Program Apollo: 1961-1969
- Konkurs na projekt cyfrowego autopilota LM - 1963
- Realizacja projektu: MIT Instrumentation Laboratories 1963-1969 [Richard J. Gran]

Moduł lądowania na księżycu



- Brak opracowań optymalnych procedur numerycznych kontrolnych przed rokiem 1966 - ręczne kodowanie i liczenie
- Ograniczona prędkość i pojemność komputera (IBM 7090)
- Brak przyjaznego środowiska graficznego do testowania, system perforowanych kart (3 osoby przez 3 lata testowały kod !)

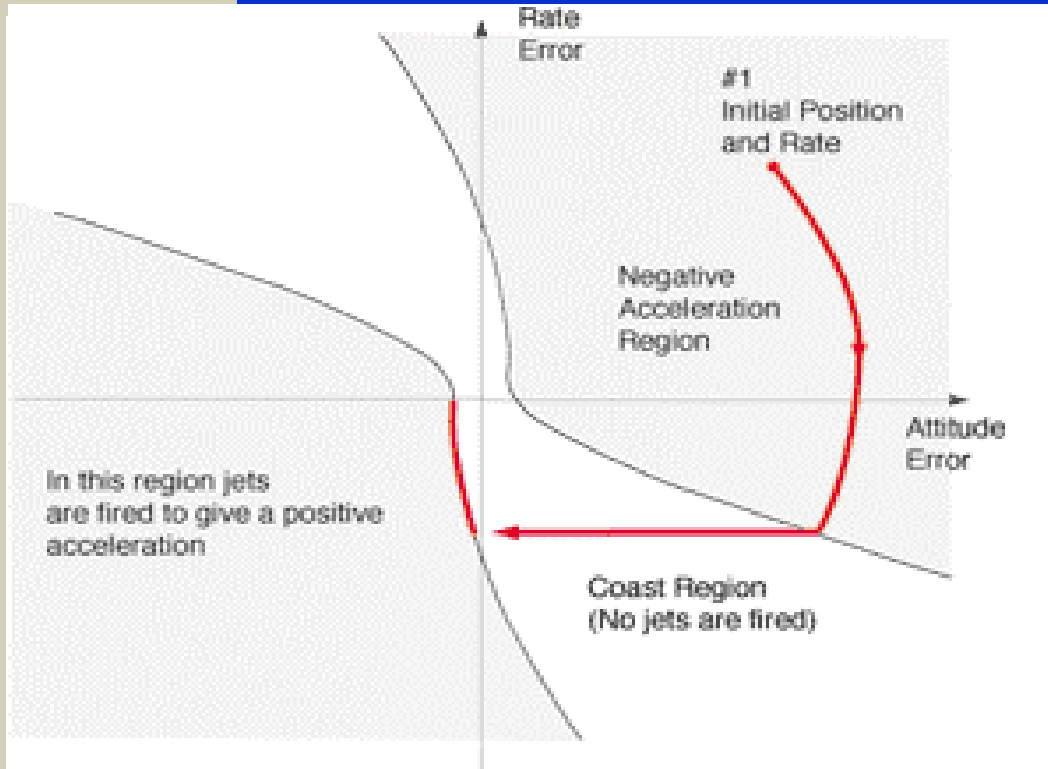
Moduł lądowania na księżycu

OP Code	0	1	2	3	4	5	6	7	Location
Extend=0		CCS	DAS			INDEX			0 1777
	TC	TCF	LXCH	CA	CS	DXCH	AD	MASK	2000 3777
			NCR			TS			4000 5777
			ADS			XCH			6000 7777
Extend=1	Read/Wite	DV	MSU				SU		0 1777
	Rand/Wand	BZF	QXCH	DCA	DCS	INDEX	BZMF	MP	2000 3777
	Ror/Wor		AUG						4000 5777
	Rxor		DIM						6000 7777



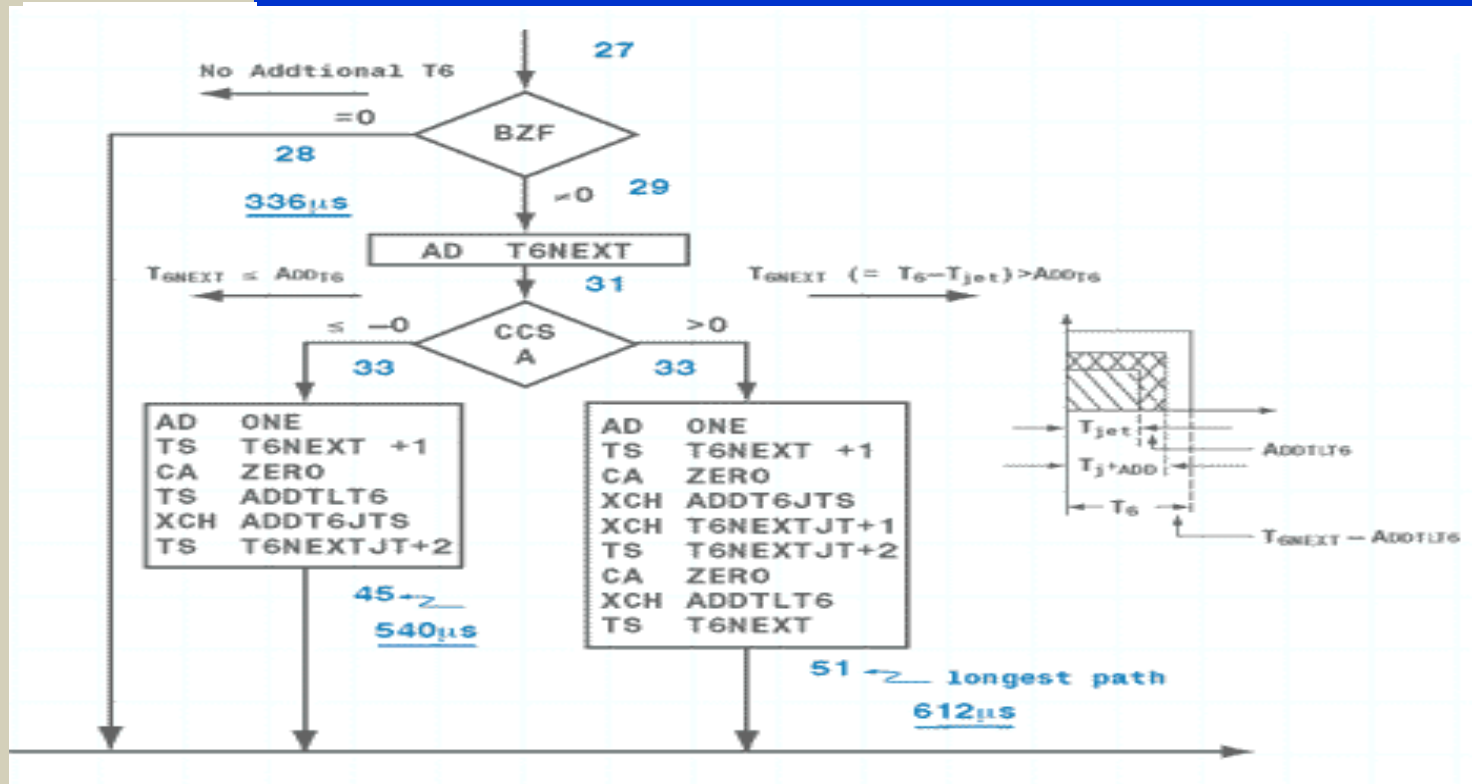
Oryginalny zestaw kodów operacyjnych dla autopilota
(implementowany na maszynie o zegarze 500 kHz !!)

Moduł lądowania na księżycu



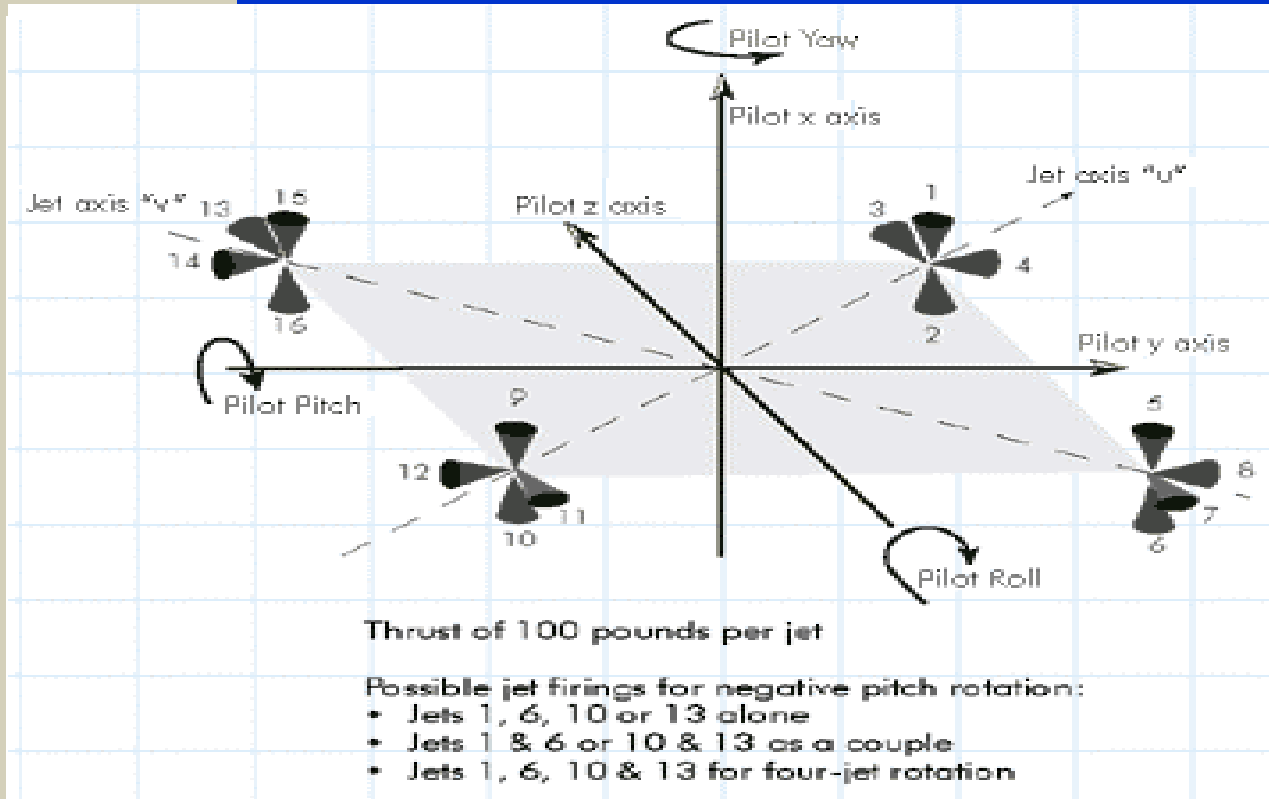
Oryginalny diagram logiczny systemu przełączania...

Moduł lądowania na księżycu



Fragment schematu blokowego – testy w Fortranie...

Moduł lądowania na księżycu



Uwzględnianie złożonych rotacji układu testowanego...

Moduł autopilot w Simulink

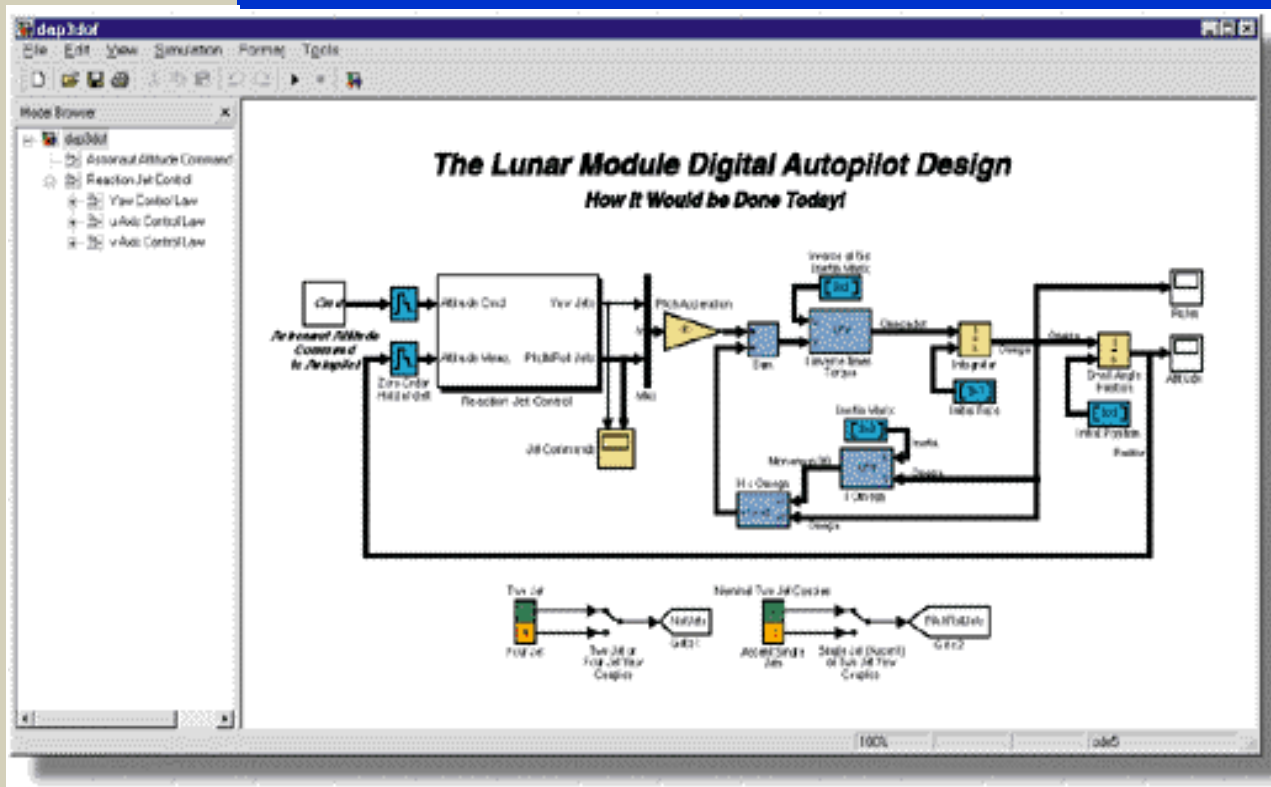
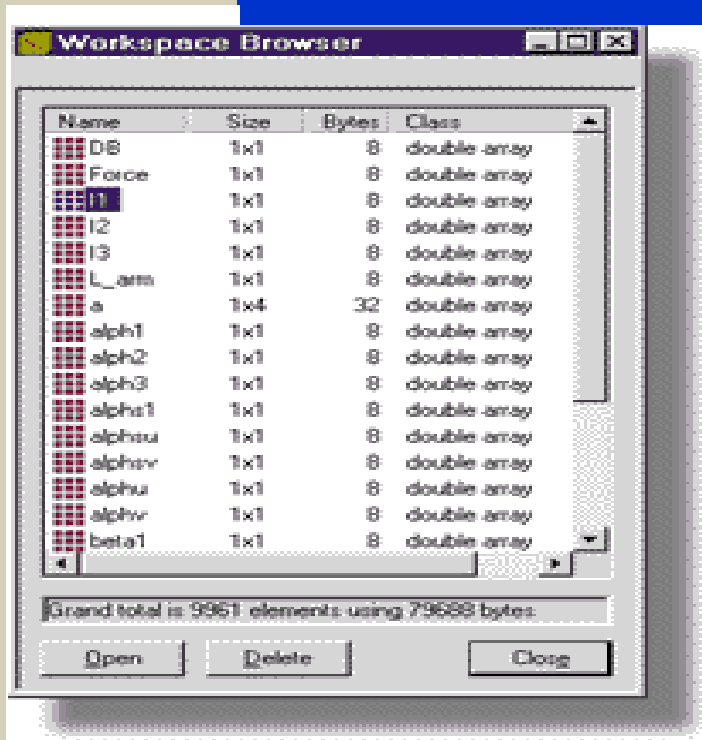


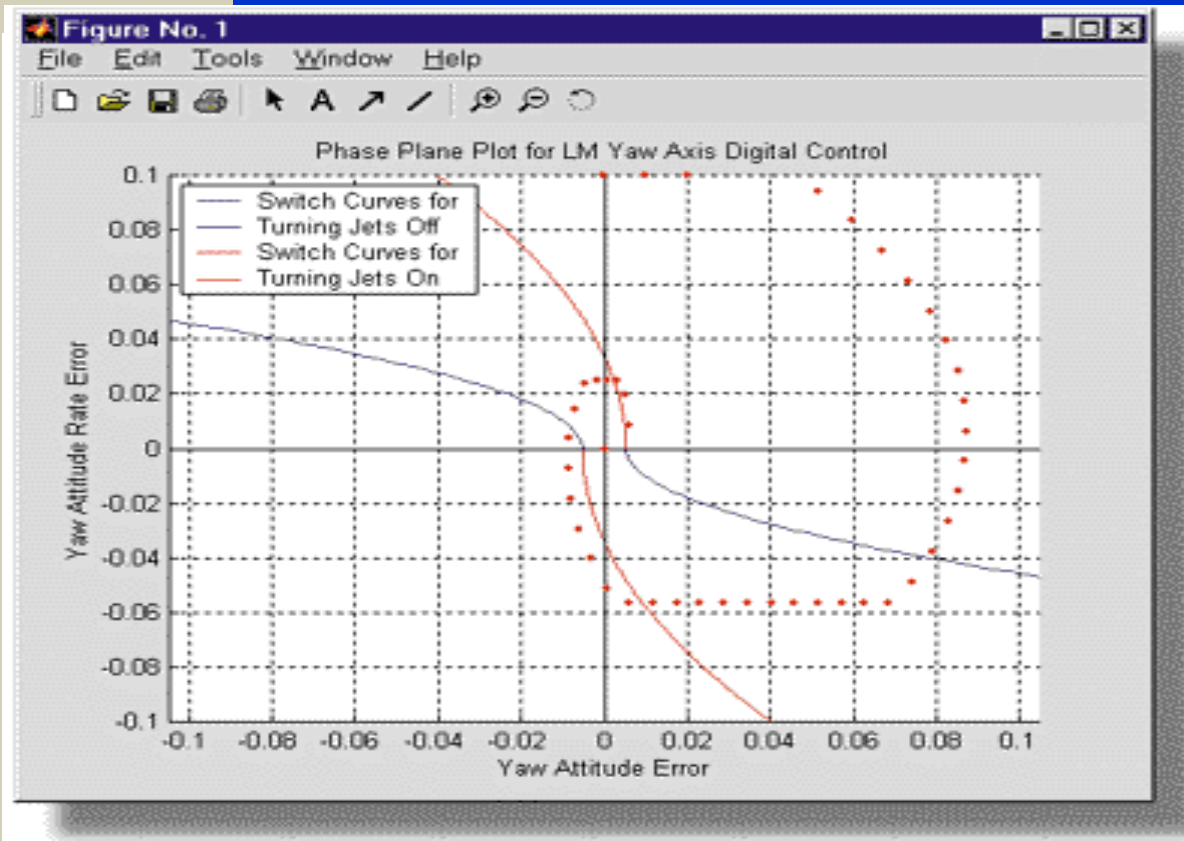
Diagram tworzony w czasie około 1 godziny...

Moduł autopilot w Simulink



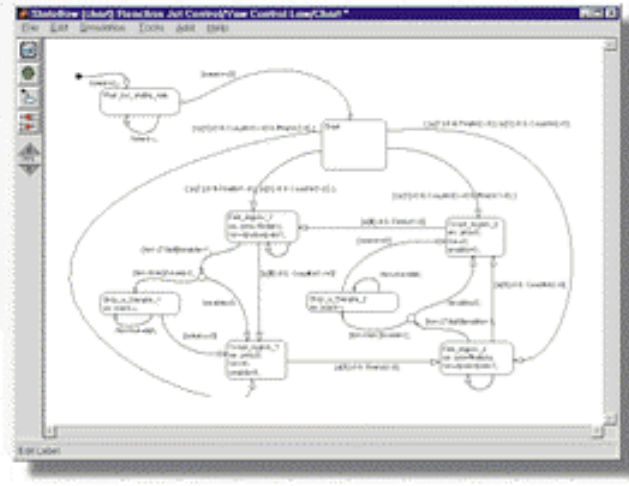
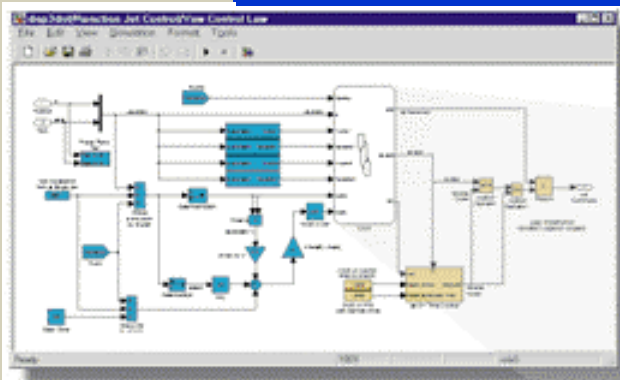
Wszystkie parametry dostępne w środowisku graficznym..

Moduł autopilot w Simulink



Możliwość interakcyjnych testów graficznych...

Moduł autopilot w Simulink



Algorytm układu kontrolnego uwzględniający złożone rotacje i odpowiedni diagram przepływów...

Moduł autopilot w Simulink



```
AdaGIDE - [jet_time_counter.adb]
File Edit Run Tools Window Help
[Icons] R 10
-- model step function
procedure Step is
begin
  -- Enable SubSystem Block: <Root>/Jet On Time Counter
  if (RCS.Enable) then

    -- Outputs for enable system: <Root>/Jet On Time Counter
    declare
      -- local block i/o variables
      RT_B_Temp2 : THW_Types.Real_T;

    begin

      -- Sum Block: <S1>/Sum1
      RT_B_Temp2 := RCS.Tic - RCS.Tic_Ts;

      -- Sum Block: <S1>/Sum
      RT_B_Temp2 := RCS.Ton - RT_B_Temp2;

      -- RelationalOperator Block: <S1>/Relational Operator
      Stop := (RT_B_Temp2 > RT_C_S1_Constant);

    end;

  end if;

  -- Output Block: <Root>/stop
  RT_V.Root_Stop := Stop;

  -- (no update code required)
end Step;
```

Fragment generowanego kodu w języku Ada...