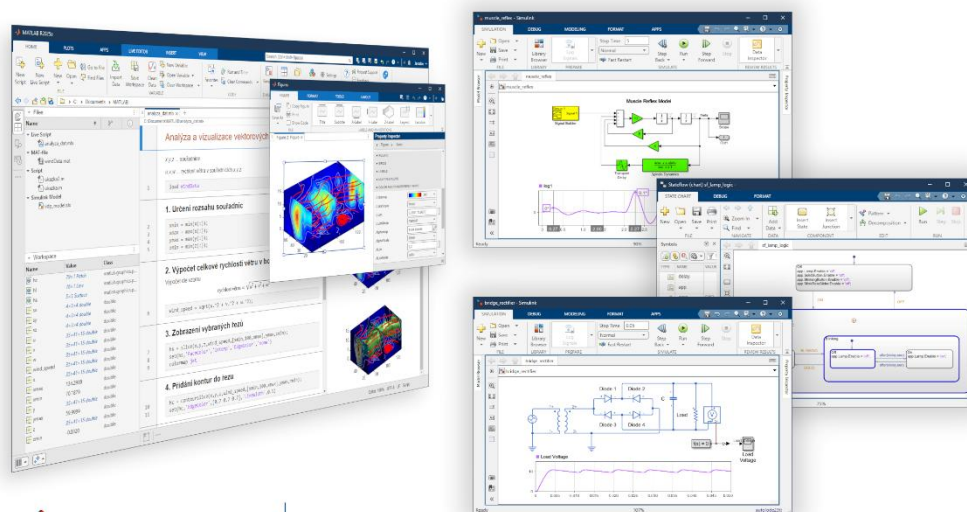


Modelovanie a simulácia nadriadenej riadiacej logiky



Michal Blaho

blaho@humusoft.sk

www.humusoft.cz

info@humusoft.cz

www.mathworks.com

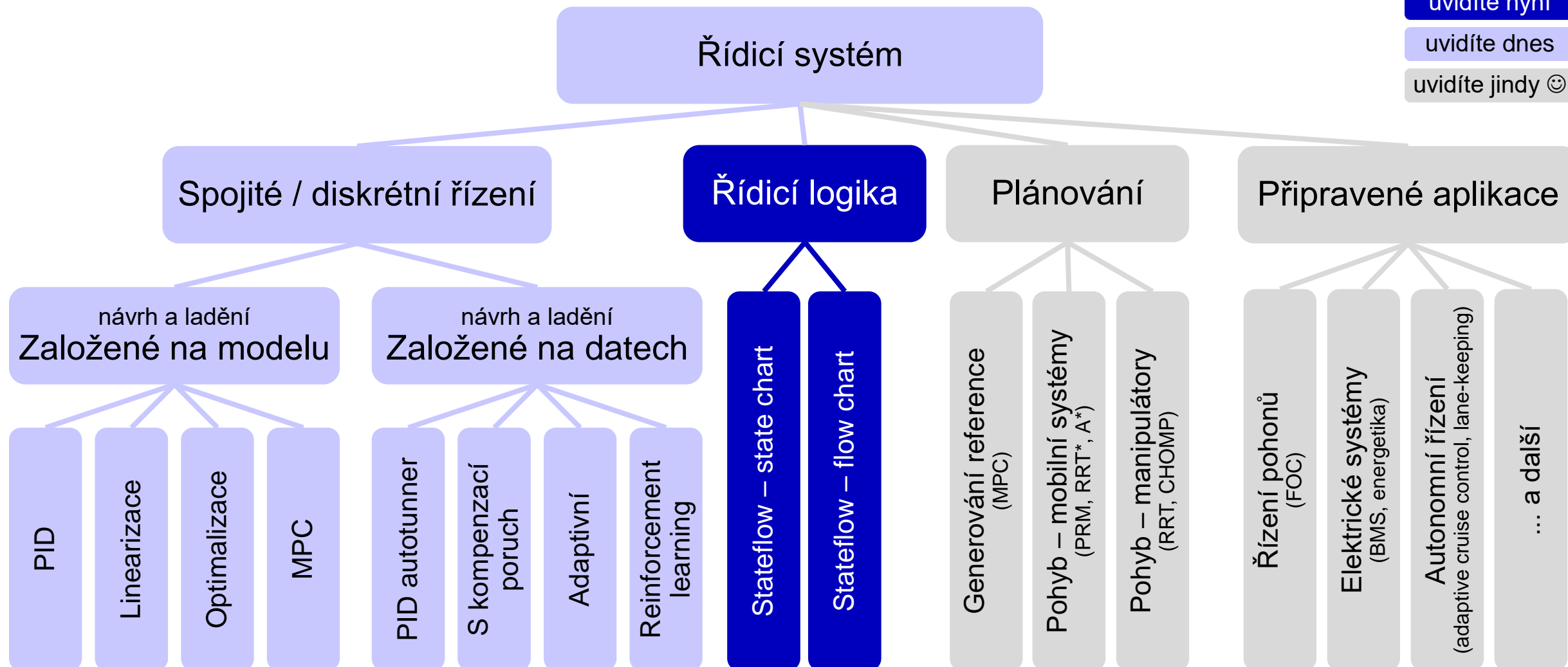
Možnosti návrhu řídicích systémů v prostředí MATLAB

(výběr možností)

uvidíte nyní

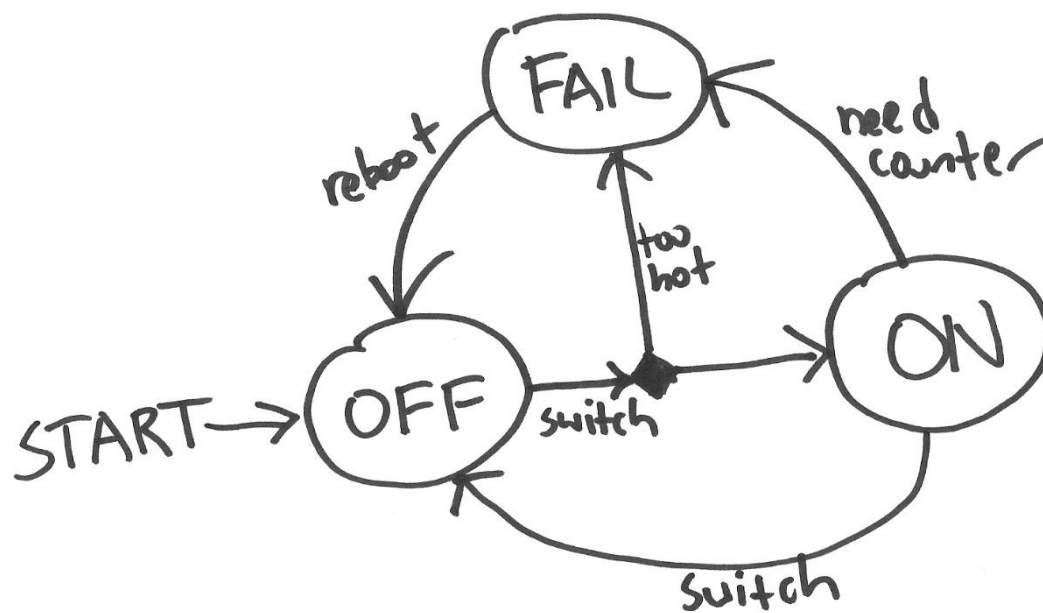
uvidíte dnes

uvidíte jindy 😊



Čo je stavový automat

- Modelovanie rôznych stavov, v ktorých sa môže systém nachádzať a ako sa prechádza medzi týmito stavmi na základe vstupov alebo udalostí



Stavové automaty sú všade



Riadenie dopravy



Automobily



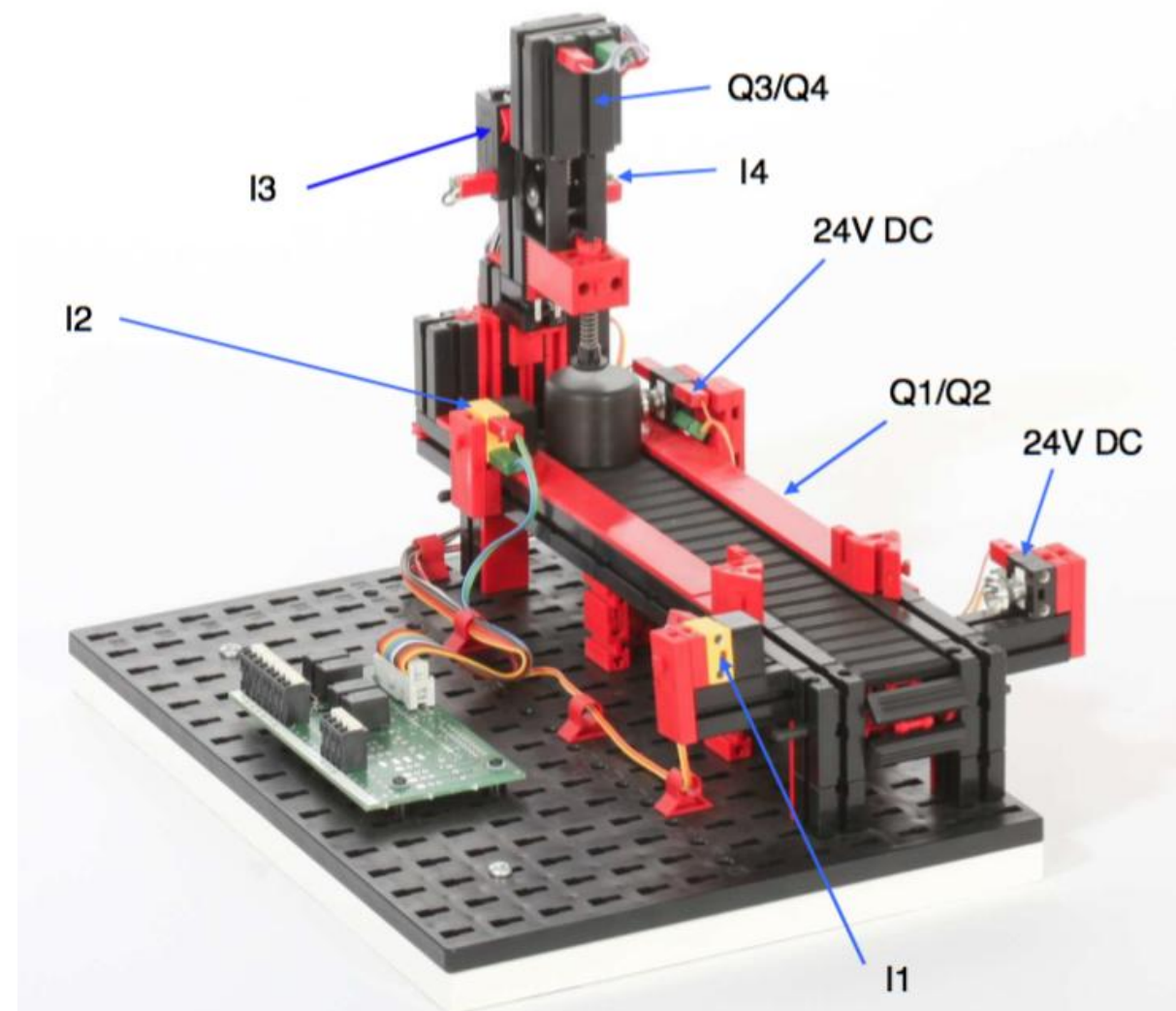
Letectvo



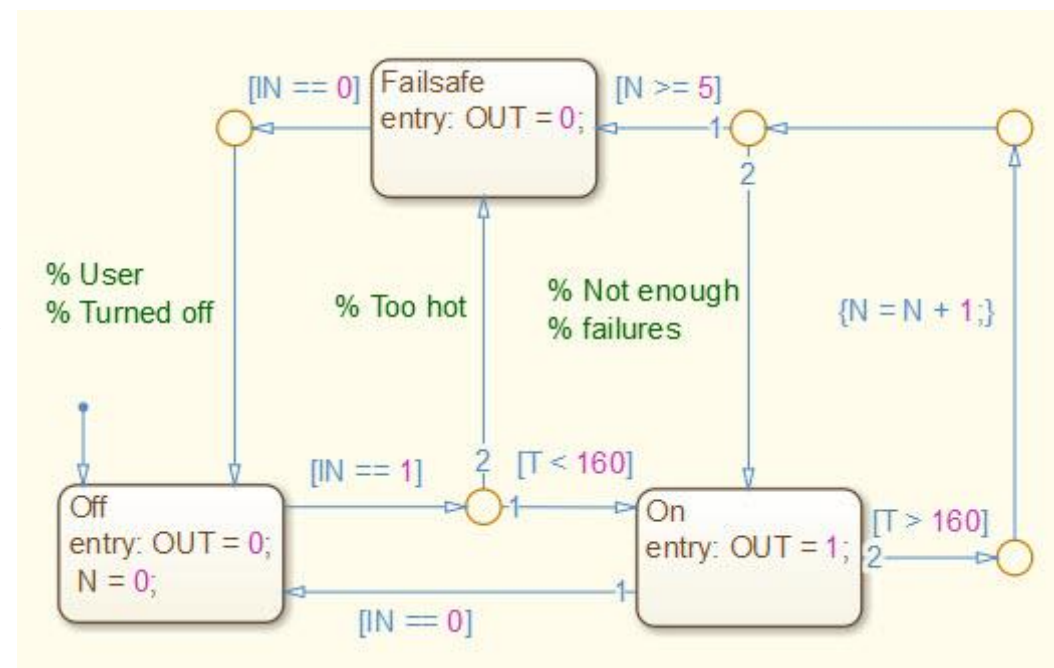
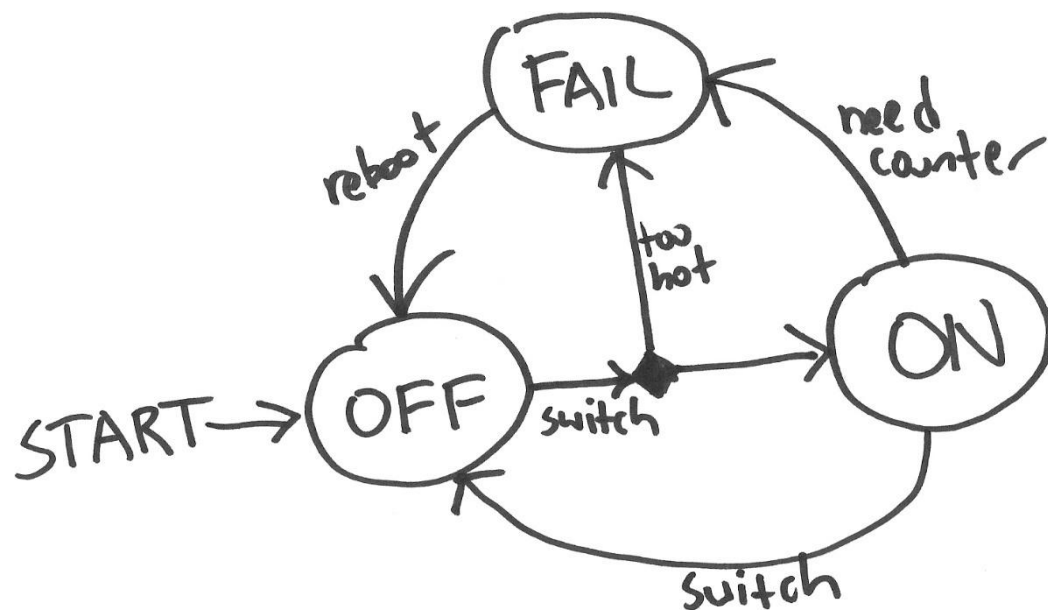
Výrobné systémy

Ukážka: Model výrobnéj linky

- Jednoduchý výrobný proces
 - posúvanie výrobku po dopravníku
 - opracovanie výrobku portálom
- Motory
 - pohyb dopravníka
 - pohyb portálu
- Snímače
 - kraje dopravníka
 - kraje portálu
- Cieľ
 - tvorba riadenia – stavový automat



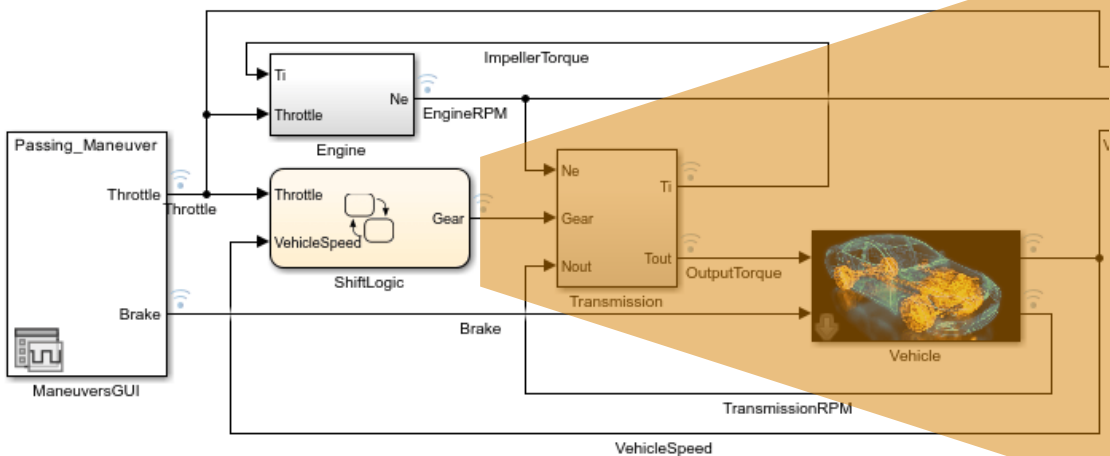
Stateflow je prostredie pre návrh stavových automatov



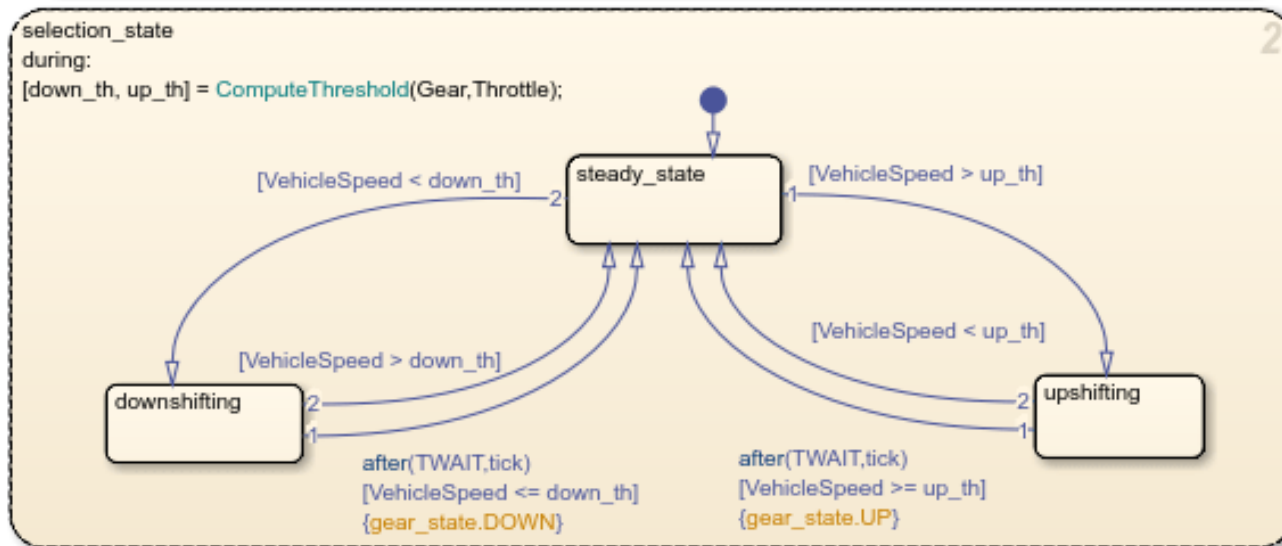
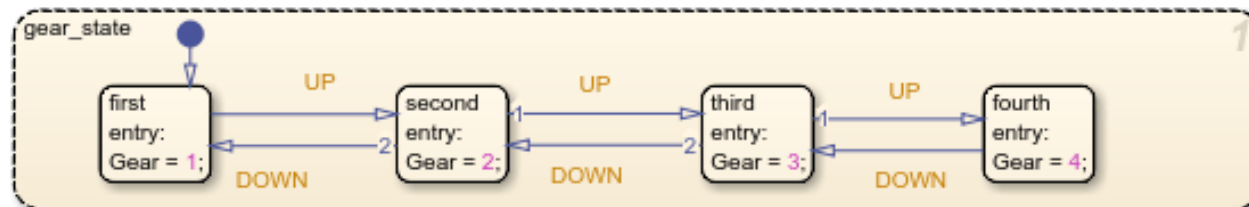
Prečo tvoriť stavové automaty
v prostredí Stateflow?

Integrácia so Simulinkom

Modeling an Automatic Transmission Controller



Copyright 1990-2024 The MathWorks, Inc.



```

Simulink Function
[down_th,up_th] = ComputeThreshold(Gear,Throttle)

```

Vizuálne modelovanie v intuitívnom prostredí

Stateflow (chart) sldemo_autotrans/ShiftLogic - Simulink

Panel nástrojov

Paleta objektov

Pracovná plocha

Vlastnosti

Symboly

The diagram illustrates a gear shifting logic. It starts with a **gear_state** containing four entry states: **first entry: Gear = 1;**, **second entry: Gear = 2;**, **third entry: Gear = 3;**, and **fourth entry: Gear = 4;**. Transitions between these states are labeled **UP** and **DOWN**. A **selection_state** is active during the process, with a **steady_state** and two sub-states: **downshifting** and **upshifting**. Transitions are triggered by **VehicleSpeed** comparisons against **down_th** and **up_th** thresholds. A **Simulink Function** block defines the threshold calculation: `[down_th, up_th] = ComputeThreshold(Gear, Throttle);`

Property Inspector

VehicleSpeed

Properties Info

Scope: Input

Port: 2

Size: Scalar

Type: double

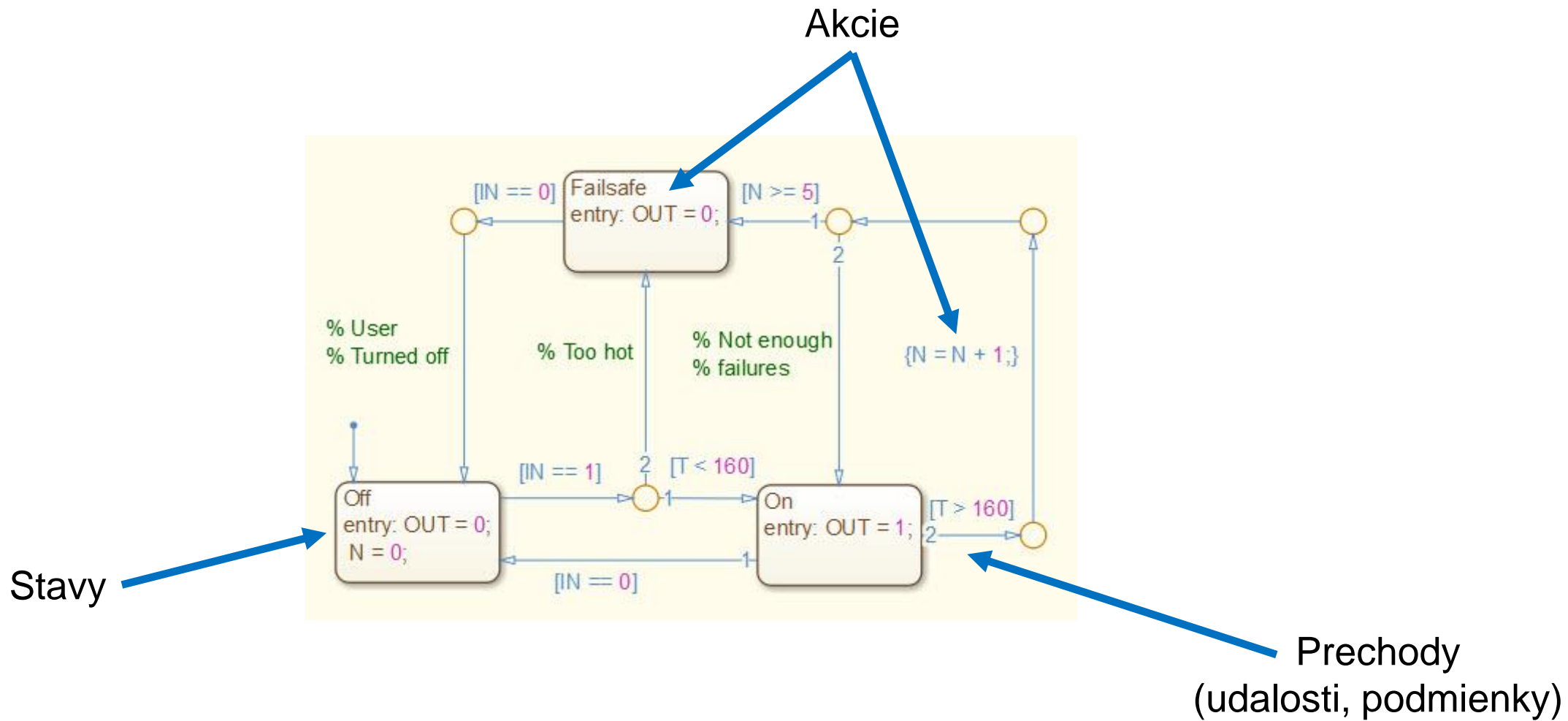
First index: Scalar

Advanced

Symbols

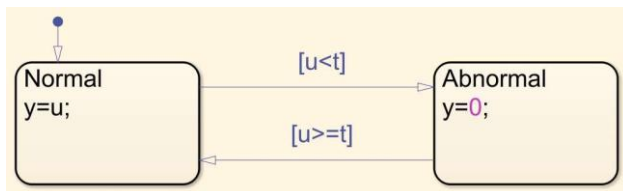
TYPE	NAME	VALUE	PORT
	ComputeThreshold		
	Throttle		1
	VehicleSpeed		2
	Gear		1
	TWAIT		
	down_th		
	up_th		
	gear_state		
	DOWN		
	UP		

Prehľadné modelovanie stavového automatu



Stateflow zostáva kompaktný

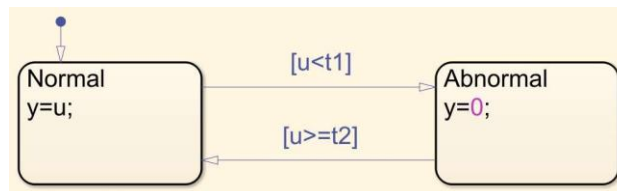
jednoduchá logika



```

for i = 1:length(inData)
    if(inData(i) >= t)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
    
```

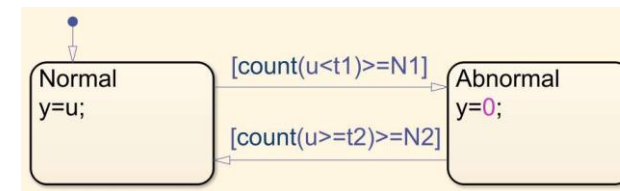
+ hysteréza



```

inNormalRegion = true;
for i = 1:length(inData)
    if(inNormalRegion && (inData(i)<t1))
        inNormalRegion = false;
    elseif(~inNormalRegion && (inData(i)>=t2))
        inNormalRegion = true;
    end
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
    
```

+ hysteréza + ošetrovanie zákmitov

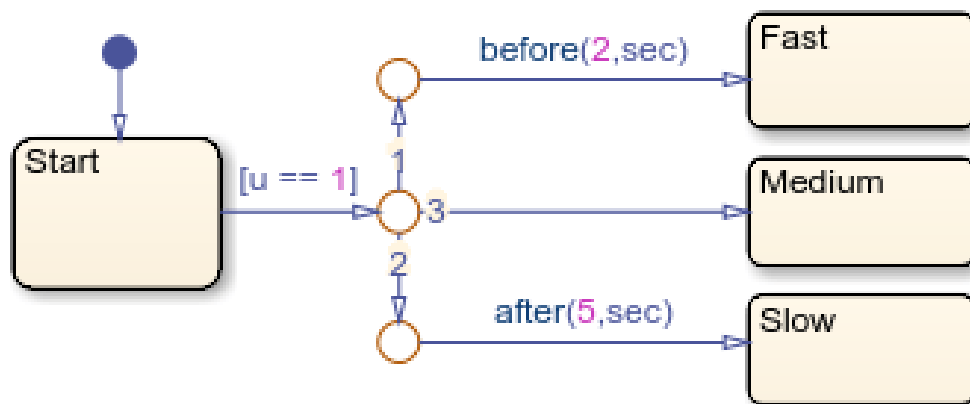


```

inNormalRegion = true;
counter = 0;
for i=1:length(inData)
    if(inNormalRegion)
        if(inData(i)<t1)
            counter = counter+1;
            if(counter>=N1)
                inNormalRegion = false;
            end
        else
            counter = 0;
        end
    else
        if(inData(i)>=t2)
            counter = counter+1;
            if(counter>=N2)
                inNormalRegion = true;
            end
        else
            counter = 0;
        end
    end
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
    
```

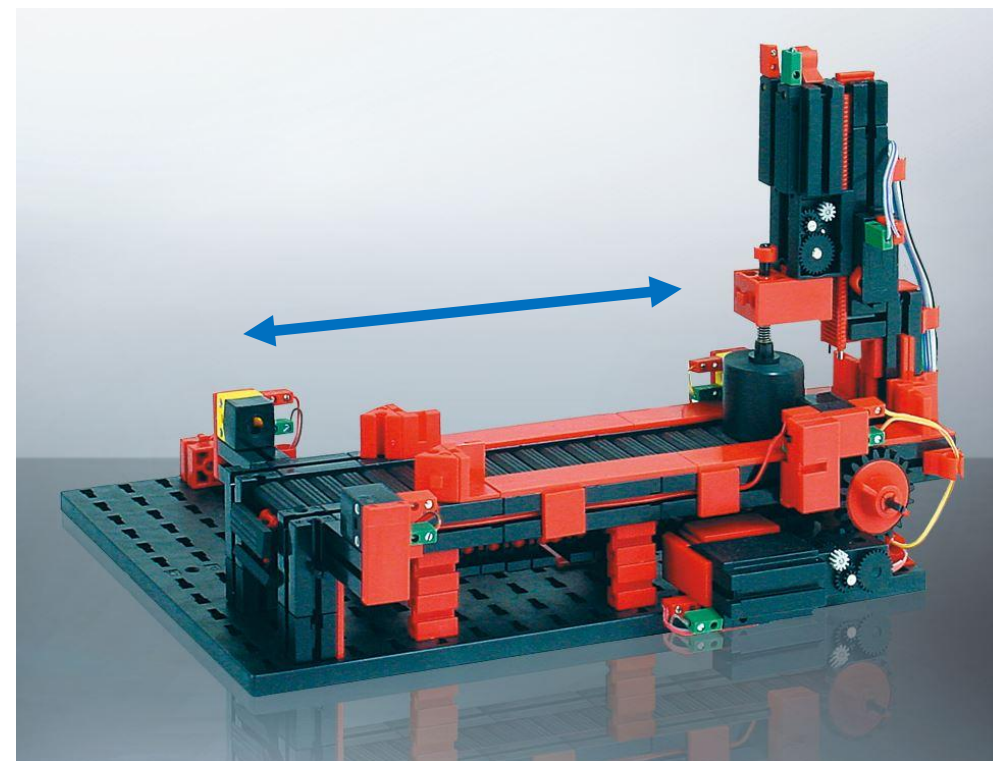
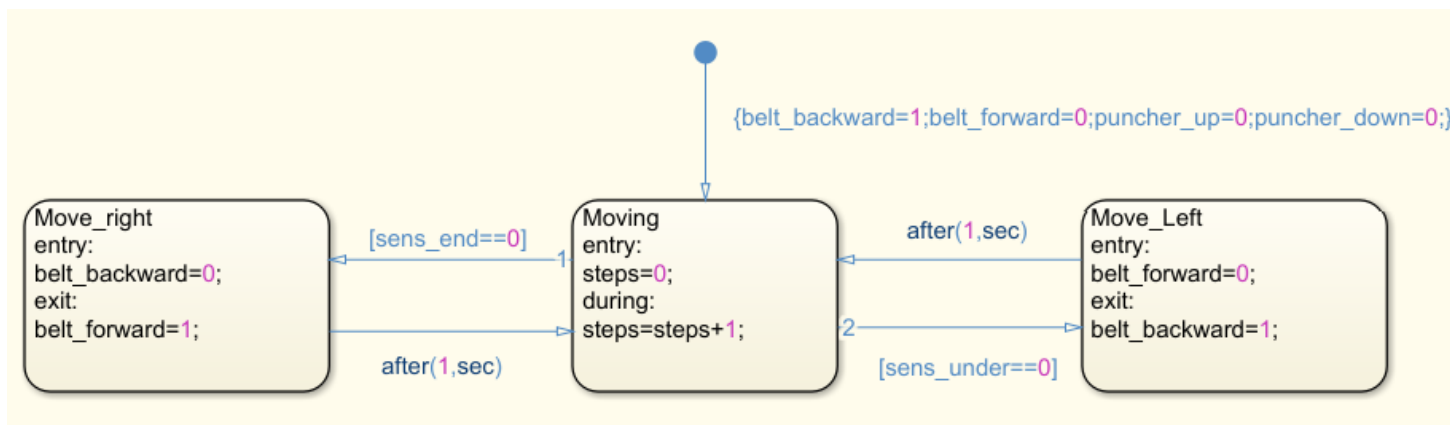
Flexibilná práca s udalosťami a časom

- Operátory – *after*, *at*, *before*, *every*, *count*, *temporalCount*, *duration*, *elapsed*
 - explicitné a implicitné udalosti – *after(n,E)*, *before(n,tick)*
 - absolútny čas (sec, msec, usec) – *after(n,sec)*

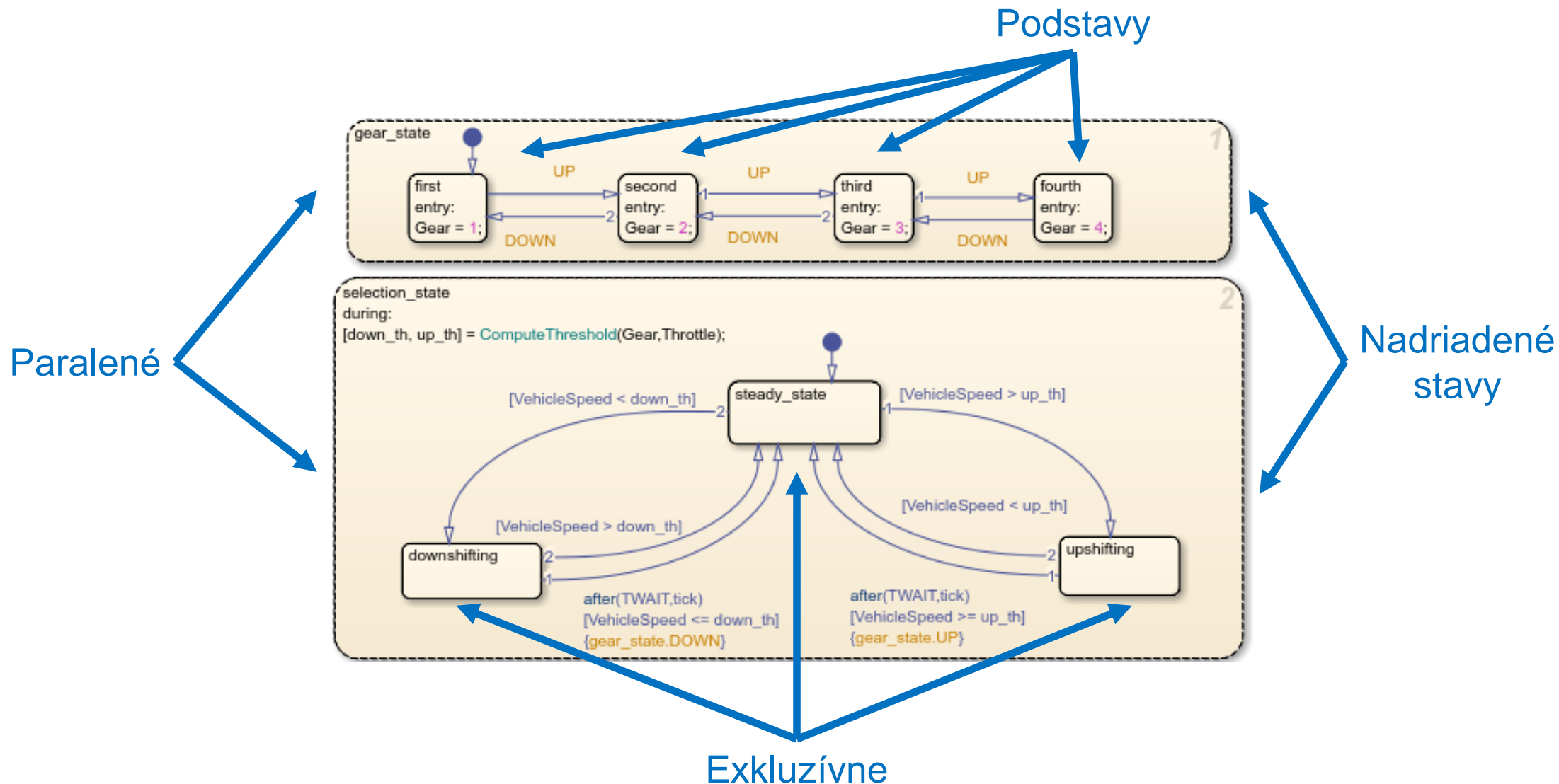


Ukážka: Základný pohyb dopravníka

- Pohyb výrobku zo strany na stranu

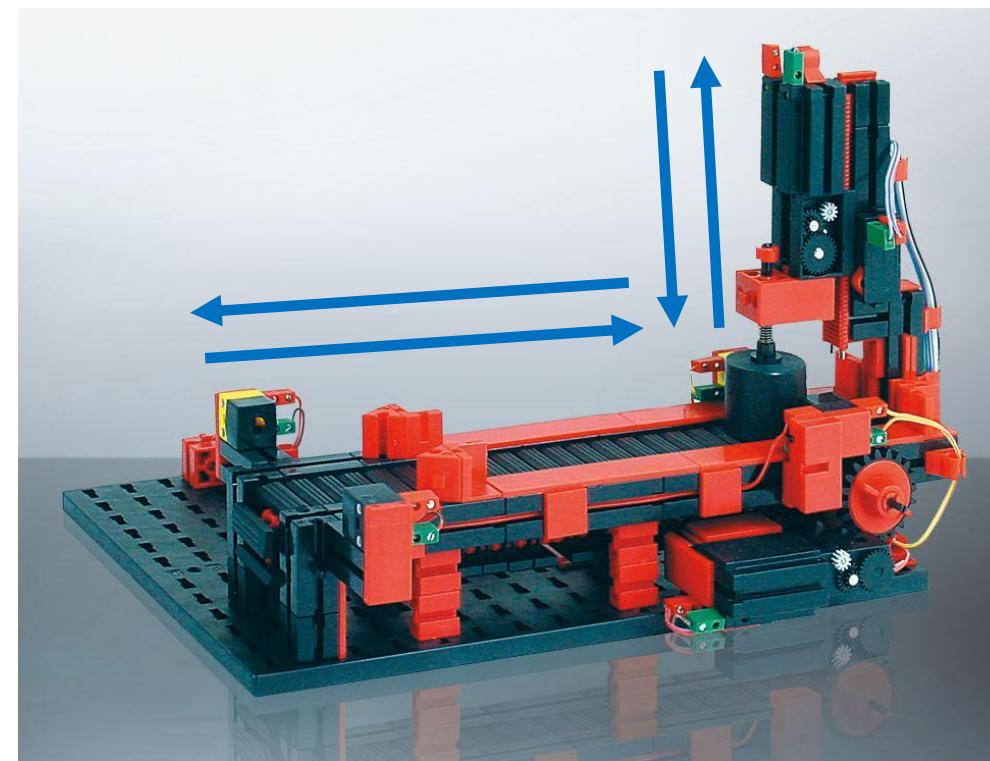
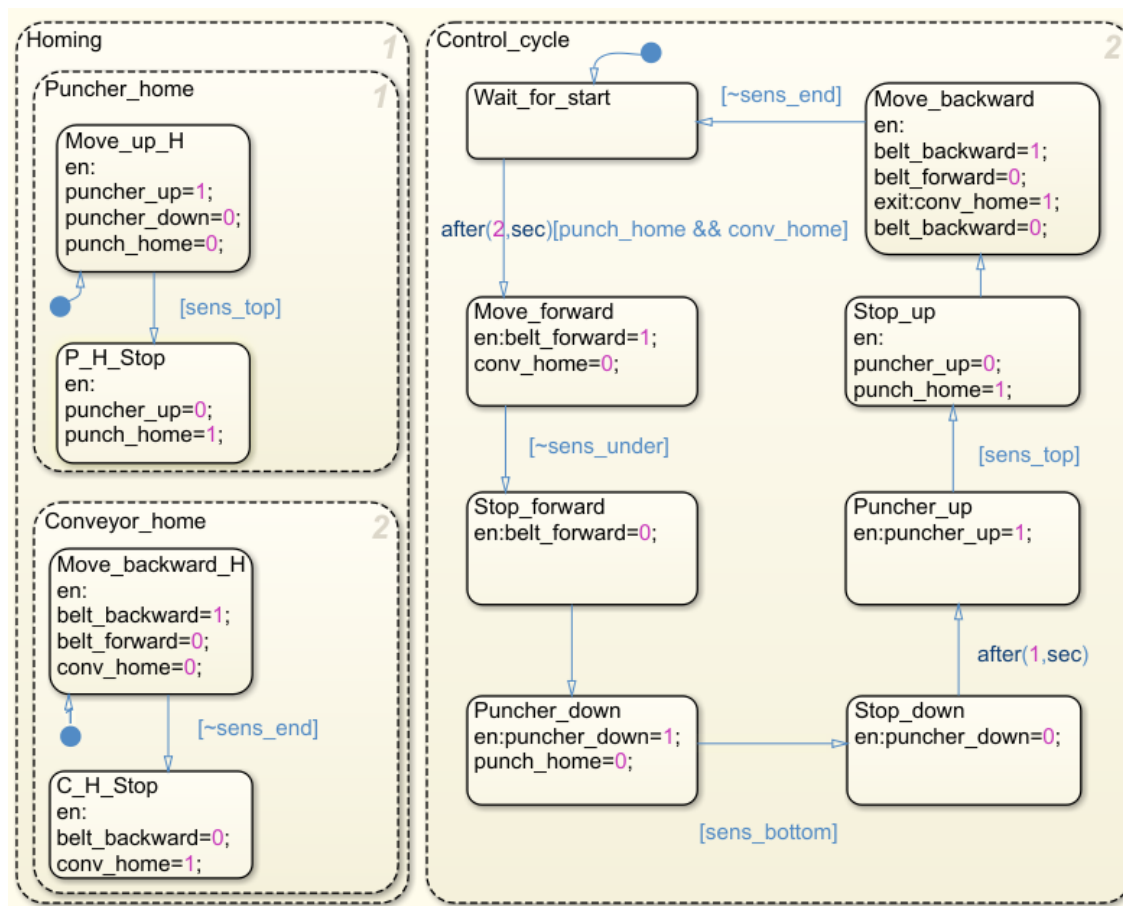


Tvorba komplexných stavových systémov

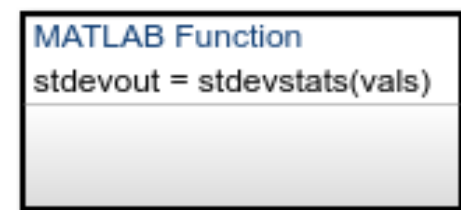


Ukážka: Komplexný pohyb dopravníka

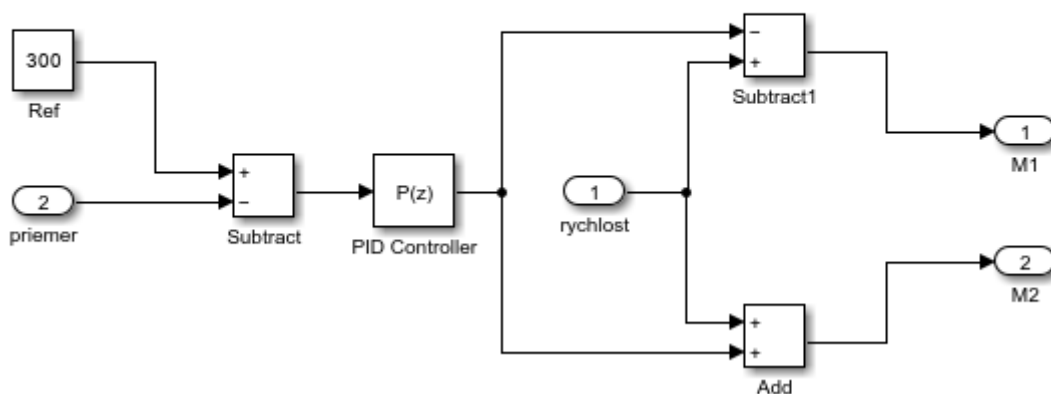
- Presun do počiatočných polôh a výrobný proces



Stavy pomocou blokov a kód prostredia MATLAB



Action



```
function stdevout = stdevstats(vals)
%#codegen

% Calculates the standard deviation for vals

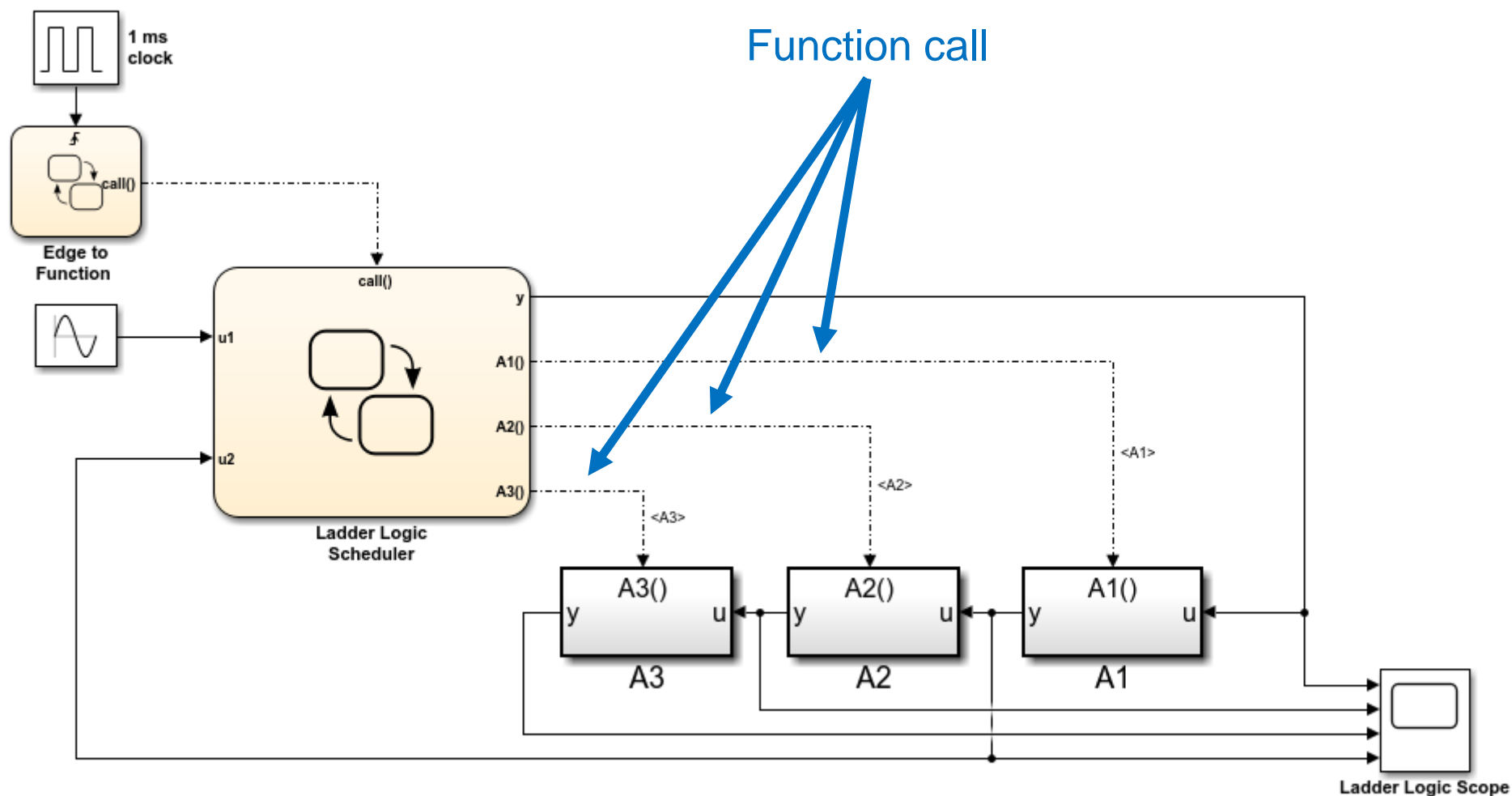
len = length(vals);
stdevout = sqrt(sum(((vals-avg(vals,len)).^2))/len);

function mean = avg(array,size)
mean = sum(array)/size;
```

Simulink State

MATLAB function

Riadenie subsystémov



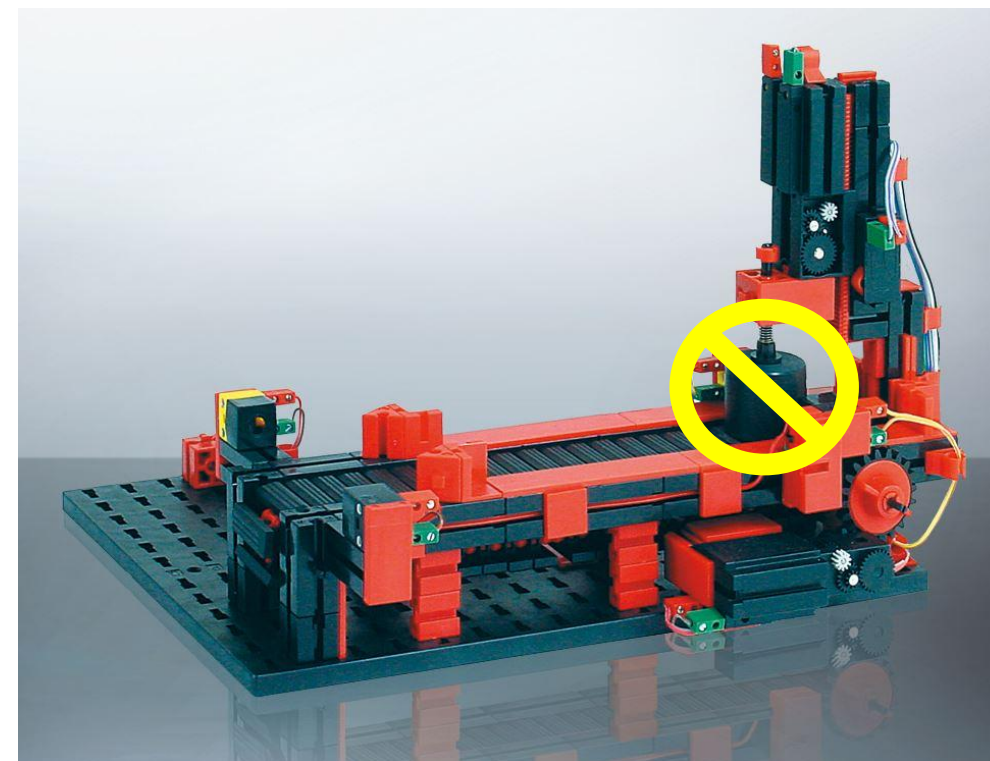
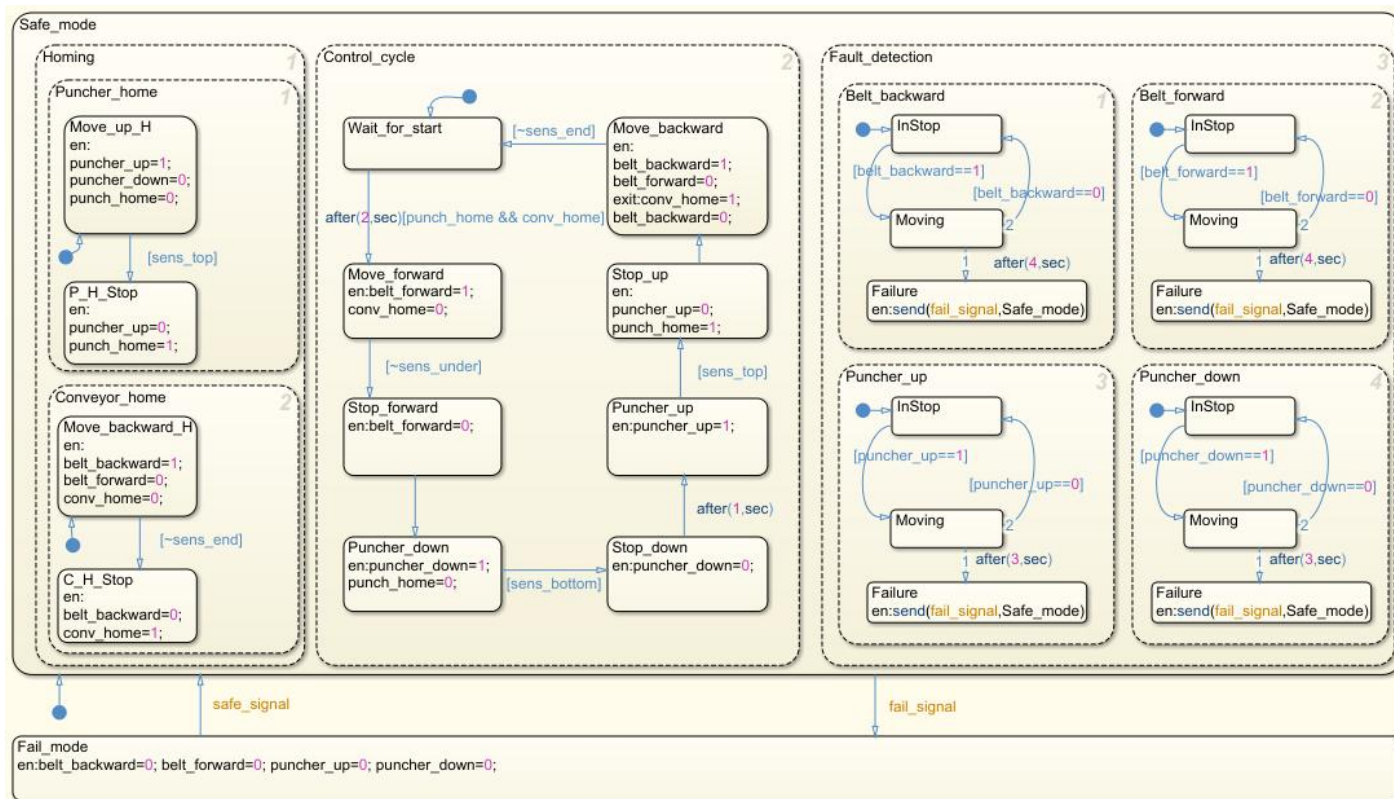
Ladenie logických chýb a nekonzistencií

Krokovanie

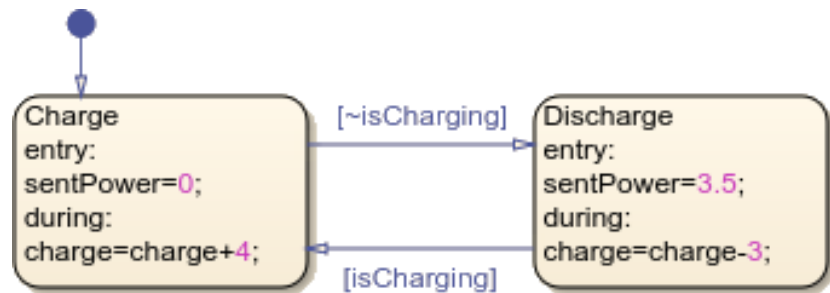
The screenshot shows the Simulink Stateflow environment. The top toolbar includes simulation controls: Step Back, Continue, Step Forward, and Stop. A blue box highlights these controls, with an arrow pointing to the text 'Krokovanie' (Stepping). The main workspace displays a state machine diagram for 'ShiftLogic'. It consists of two main state sets: 'gear_state' and 'selection_state'. 'gear_state' contains four states: 'first entry: Gear = 1;', 'second entry: Gear = 2;', 'third entry: Gear = 3;', and 'fourth entry: Gear = 4;'. Transitions between these states are triggered by 'UP' and 'DOWN' events. 'selection_state' contains three states: 'steady_state', 'downshifting', and 'upshifting'. Transitions are based on conditions like '[VehicleSpeed < down_th]' and '[VehicleSpeed > up_th]'. A blue box highlights the 'steady_state' in 'selection_state', with an arrow pointing to the text 'Breakpoint'. Another blue box highlights the 'first entry' state in 'gear_state', with an arrow pointing to the text 'Aktívne stavy' (Active states). At the bottom, a Simulink Function block is defined as: `[down_th, up_th] = ComputeThreshold(Gear, Throttle)`. The status bar at the bottom indicates 'During: State steady_state', 'View diagnostics', '148%', 'T=0.200', '0%', and 'ode5'.

Ukážka: Reakcia na poruchu

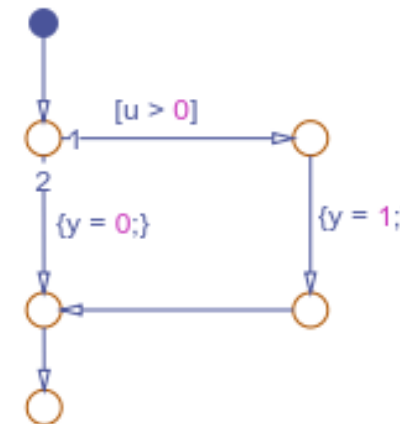
- Chýbajúci alebo zaseknutý výrobok



Viaceré prístupy modelovania



Chart



Flow chart

Condition Table						
	DESCRIPTION	CONDITION	D1	D2	D3	D4
1	x is equal to 1	XEQ1: x == 1	T	F	F	-
2	y is equal to 1	YEQ1: y == 1	F	T	F	-
3	z is equal to 1	ZEQ1: z == 1	F	F	T	-
ACTIONS: SPECIFY A ROW FROM THE ACTION TABLE			A1	1	A2	2

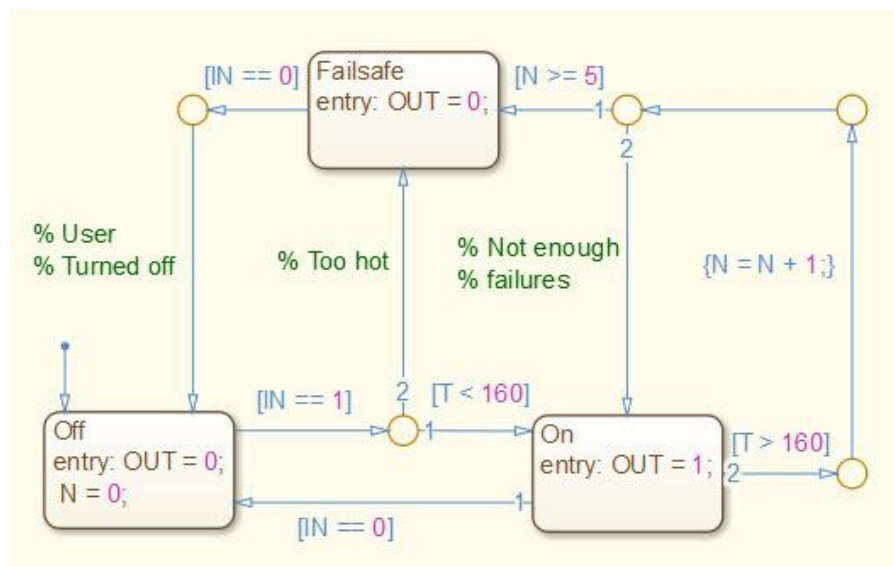
Action Table		
	DESCRIPTION	ACTION
1	set r to 1	A1: r=1;
2	set r to 2	A2: r=2;

Truth Table

STATES	TRANSITIONS	
	IF	ELSE-IF(2)
ON en: y = 1;	[u == 0]	
ON_TO_OFF		
ON_TO_OFF	after(0.1,sec)	[u == 1]
OFF_TO_ON	after(0.1,sec)	[u == 0]
OFF_TO_ON		
OFF en: y = 0;	[u == 1]	
OFF_TO_ON		

State Transition Table

Rýchly prechod od návrhu k nasadeniu

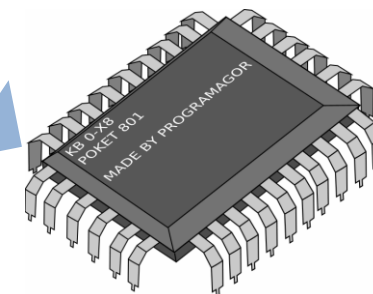


```

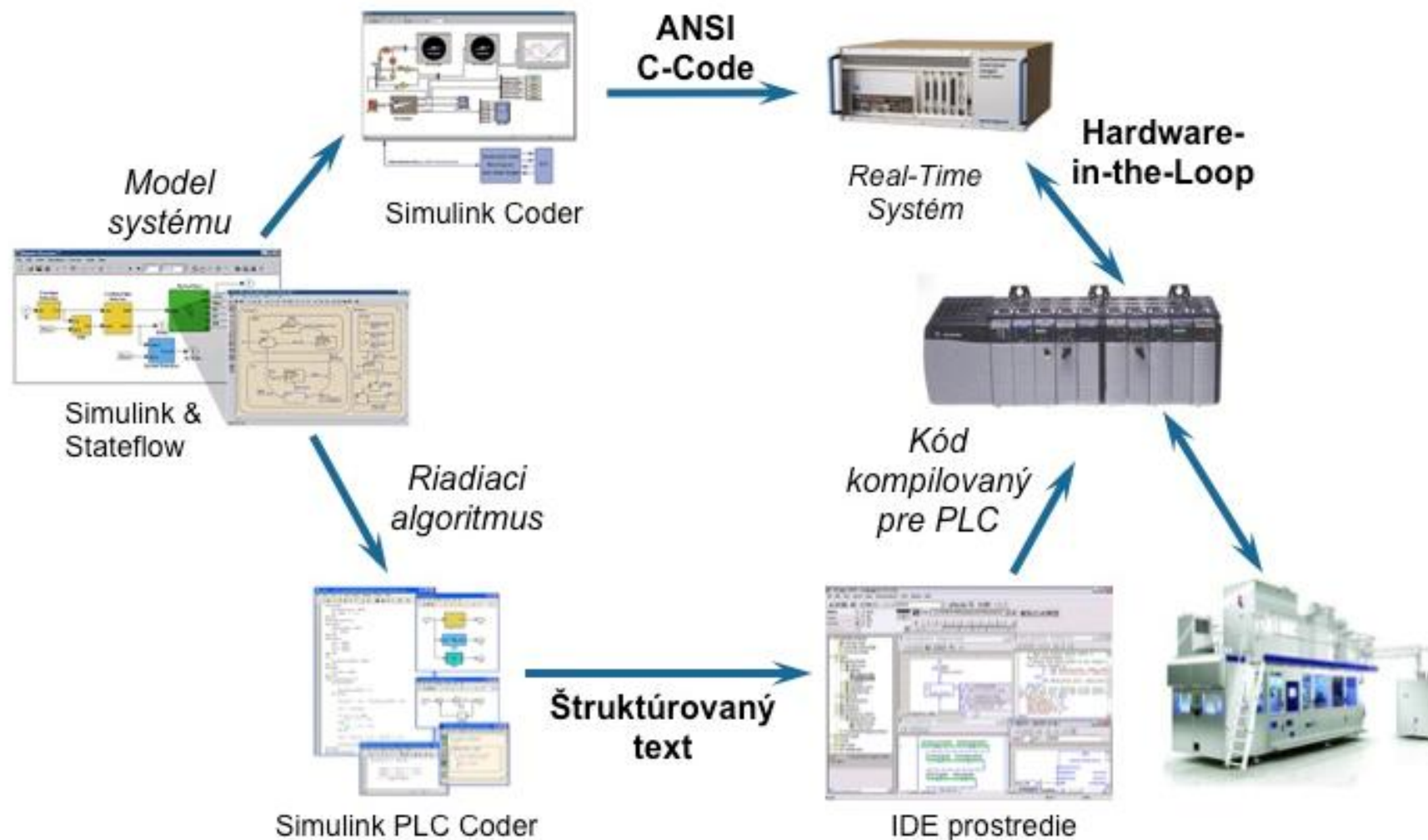
/* Model step function */
void Chart_step(void)
{
  /* Chart: '<Root>/Chart' incorporates:
   * Inport: '<Root>/IN'
   * Inport: '<Root>/T'
   */
  /* Gateway: Chart */
  /* During: Chart */
  if (Chart_DWork.is_active_c3_Chart == 0U) {
    /* Entry: Chart */
    Chart_DWork.is_active_c3_Chart = 1U;

    /* Entry Internal: Chart */
    /* Transition: '<S1>:137' */
    Chart_DWork.is_c3_Chart = Chart_IN_Off;

    /* Outport: '<Root>/OUT' */
    /* Entry 'Off': '<S1>:125' */
    Chart_Y.OUT = 0.0;
    Chart_DWork.N = 0.0;
  } else {
    switch (Chart_DWork.is_c3_Chart) {
      case Chart_IN_Failsafe:
        /* During 'Failsafe': '<S1>:138' */
        if (Chart_U.switch_on == 0.0) {
          /* Transition: '<S1>:129' */
          /* Transition: '<S1>:123' */
          /* User */
          /* Turned off */
          Chart_DWork.is_c3_Chart = Chart_IN_Off;
        }
      }
    }
  }
}
  
```

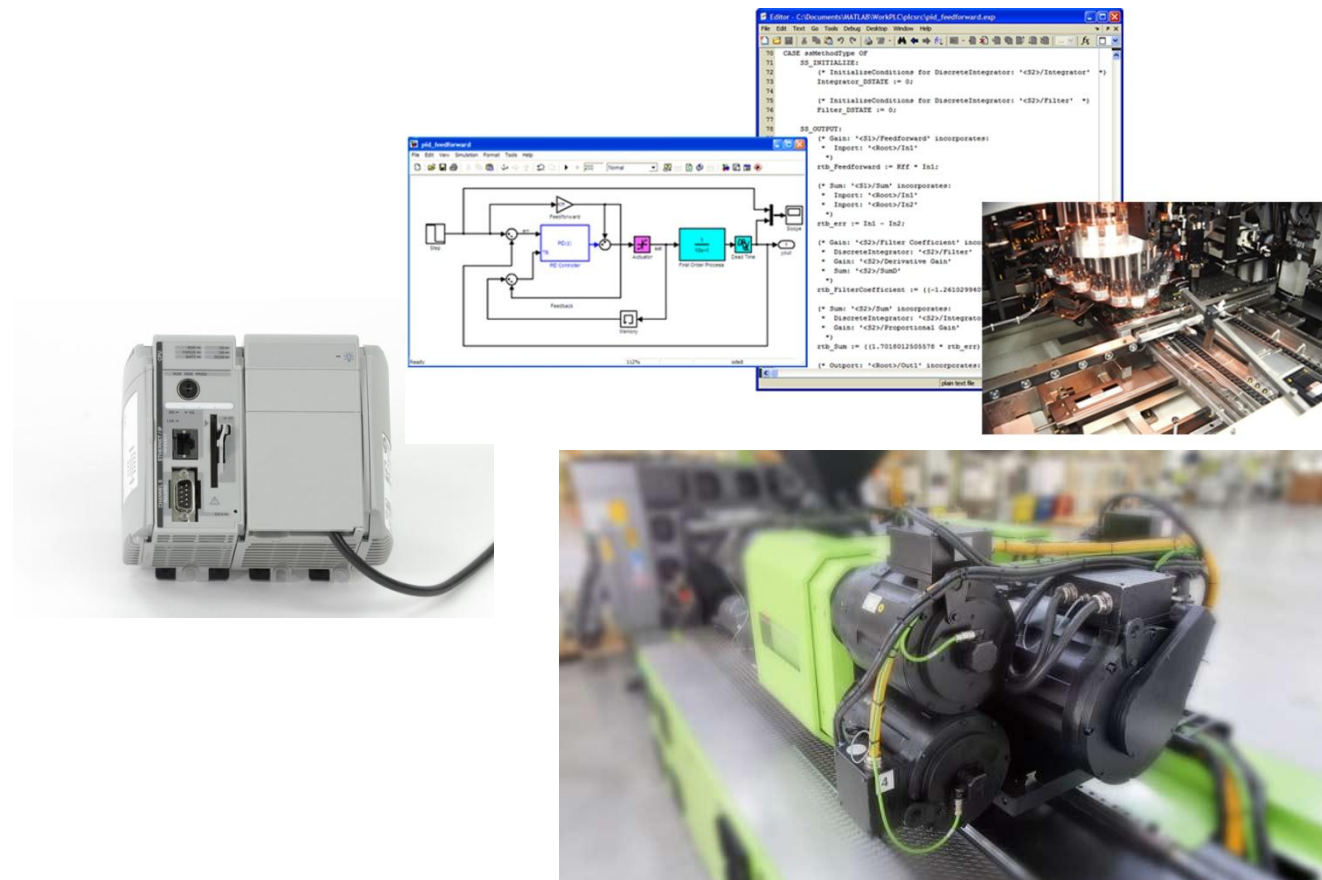


Rýchly prechod od návrhu k nasadeniu



Simulink PLC Coder

- IEC 61131-3 štruktúrovaný text
 - MATLAB
 - Simulink
 - Stateflow
- Podpora rôznych výrobcov
 - Rockwell Automation Studio 5000
 - Siemens STEP 7/TIA Portal
 - B&R Automation Studio
 - Beckhoff TwinCAT
 - PLCopen XML, Generic File Format
 - a ďalšie IDE a PLC

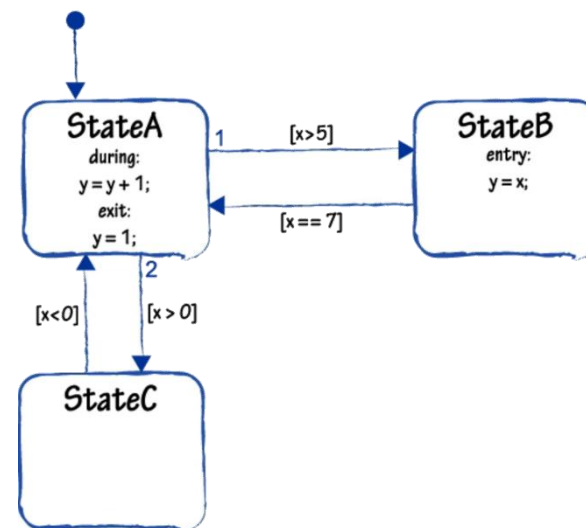


Prečo tvoriť stavové automaty v prostredí Stateflow?

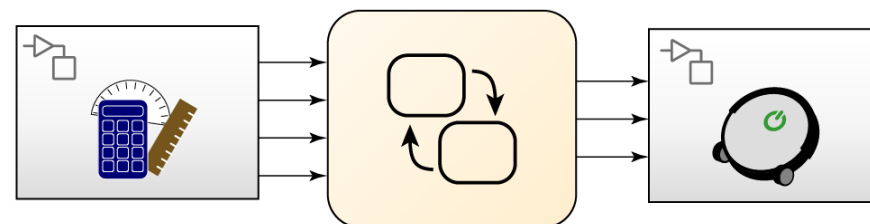
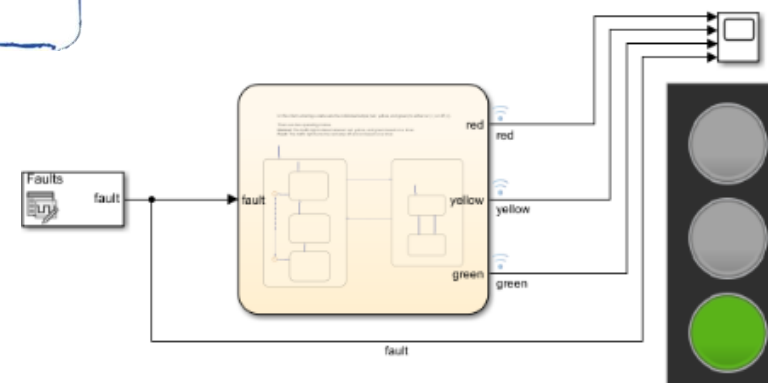
- ✓ Integrácia so Simulinkom
- ✓ Vizúálne modelovanie v intuitívnom prostredí
- ✓ Prehľadné modelovanie stavového automatu
- ✓ Kompaktnosť modelu
- ✓ Flexibilná práca s udalosťami a časom
- ✓ Tvorba komplexných stavových systémov
- ✓ Nadradená logika – riadenie subsystémov
- ✓ Ladenie logických chýb a nekonzistencií
- ✓ Rýchly prechod od návrhu k nasadeniu

Stateflow Onramp

- Bezplatný, krátky kurz základov
- Krátke videá a praktické cvičenia
- Automatizované hodnotenia a okamžitá spätná väzba
- Obsahuje témy:
 - Stavové automaty a ich tvorba
 - Akcie a ich vykonávanie
 - Flow charts
 - Funkcie v Stateflow
 - Hierarchia stavov



Stateflow Assessment
No input required
for this task



Verejné školenia

naučte sa ovládať prostredie Stateflow pod vedením našich lektorov

- Organizuje Humusoft
- Prezenčnou aj online formou
- V češtine, slovenčine, angličtine

- Ponúkame tiež **privátne školenia** na Vami vybrané témy

- Viacej informácií na **webe**:

<https://www.humusoft.cz/training/>



Ďakujem za pozornosť.

Otázky?