

Modelování polohových servomechanismů v prostředí Matlab / Simulink

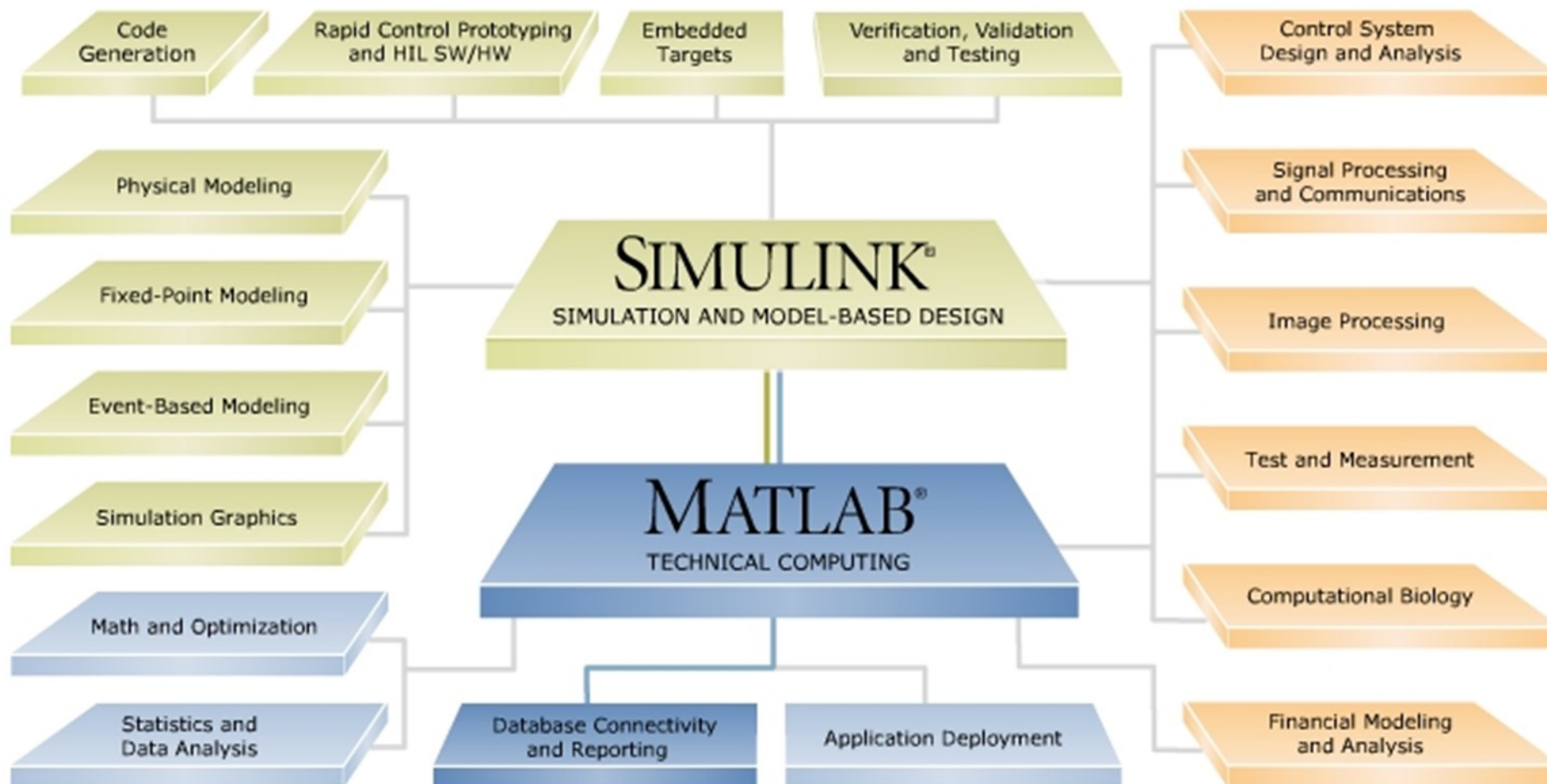
Lachman Martin, Mendřický Radomír – Elektrické pohony a servomechanismy

27.11.2013

Struktura programu MATLAB-SIMULINK

Simulink Product Family

Application-Specific Products

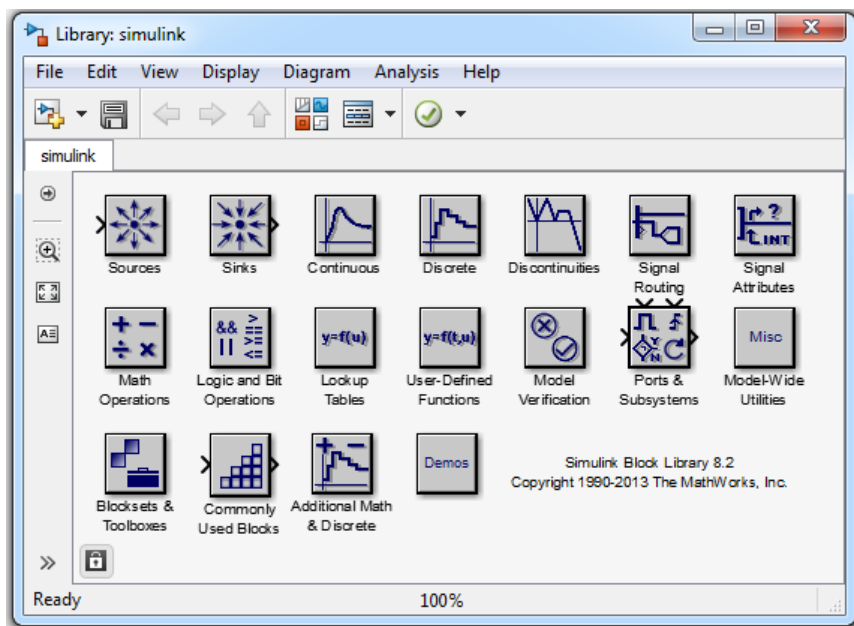


MATLAB Product Family

SIMULINK

- Pracuje pouze společně s Matlabem,
- používá se pro simulaci a modelování dynamických systémů,
- využívá algoritmy MATLABu pro numerické řešení nelineárních diferenciálních rovnic,
- poskytuje možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat a rovnic.

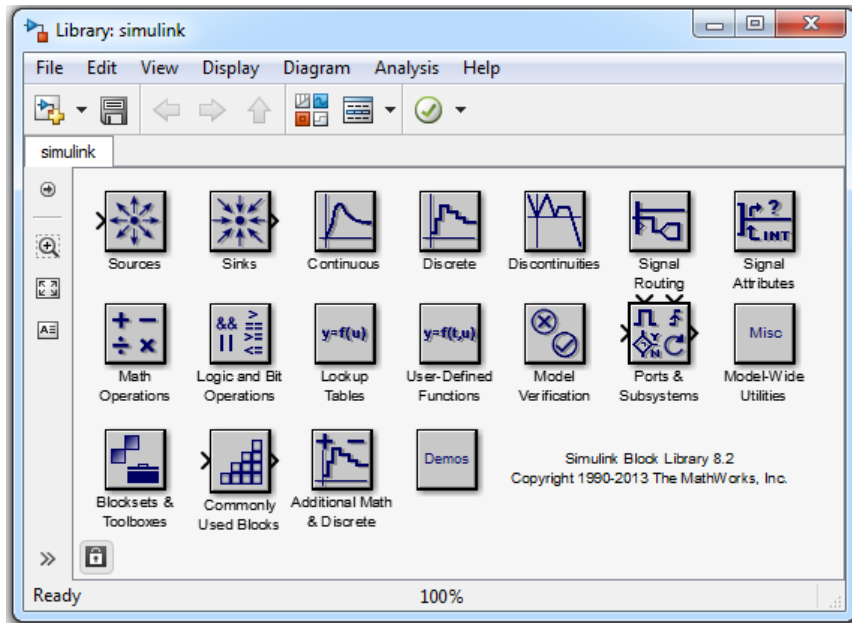
Knihovna Simulinku



Přehled jednotlivých knihoven

- **Sources** – generování signálu: *Constant, Step, Clock, Sine Wave, Signal Generator*, atd.
- **Sinks** - zobrazení a uložení signálu: *Scope, XY Graph, Display, To File, To Workspace*, atd.
- **Continuous** - modelování spojitých dějů: *Integrator, Derivative, PID Controller, Transfer Function*, atd.
- **Discrete** – modelování diskrétních dějů: *Unit Delay, Difference, Discrete PID Controller*, atd.
- **Discontinuities** – modelování nespojitých dějů: *Relay, Saturation, Dead zone*, atd.
- **Signal Routing** – práce se signály: *Mux, Demux, Manual Switch*, atd.
- **Signal Attributes** – mění se typy a vlastnosti signálu.
- **Math Operations** – matematické operace: *Sum, Gain, Product, Divide, Abs, Math Function*, atd.
- **Logic and Bit Operations** – logické a bitové operace: *Logical operator, Compare To Constant, Detect Change*, atd.

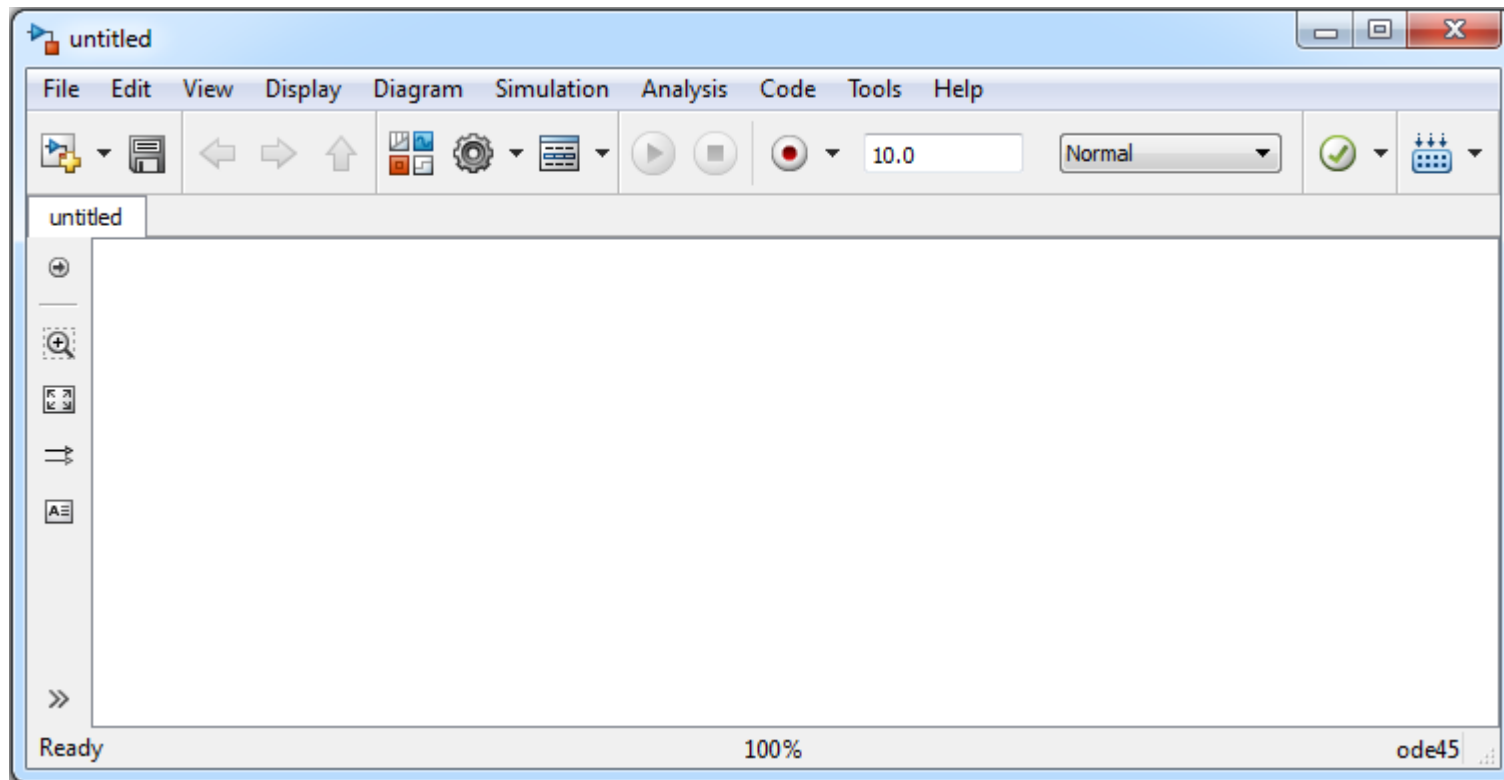
Knihovna Simulinku



Přehled jednotlivých knihoven

- **Lookup Tables** – vyhledávací tabulky: *Interpolation Using Prelookup, 2-D Lookup Table, Lookup Table Dynamic*, atd.
- **User-Defined Function** – uživatelem definované vlastní funkce použité při simulaci (S-funkce, m-funkce).
- **Model Verification** – bloky pro ověření správnosti výstupů z modelu: *Check Static Range, Check Dynamic Gap*, atd.
- **Ports & Subsystems** – bloky portů a subsystémů vytvářených v modelu: *Subsystem, Atomic Subsystem, If Action Subsystem, In, Out*, atd.
- **Model-Wide Utilities** – rozšíření o textový popis modelu: *Model Info, DocBlock*, atd.
- **Blocksets & Toolboxes** – souhrn dalších toolboxů využívajících prostředí Simulinku.
- **Commonly Used Blocks** – zde jsou z výše uvedených knihoven Simulinku vykopírované nejčastěji používané bloky a dané do jedné knihovny.
- **Additional Math & Discrete** – speciální matematické operace.
- **Demos** – demonstrační příklady práce v Simulinku.

Modelovací okno v Simulinku

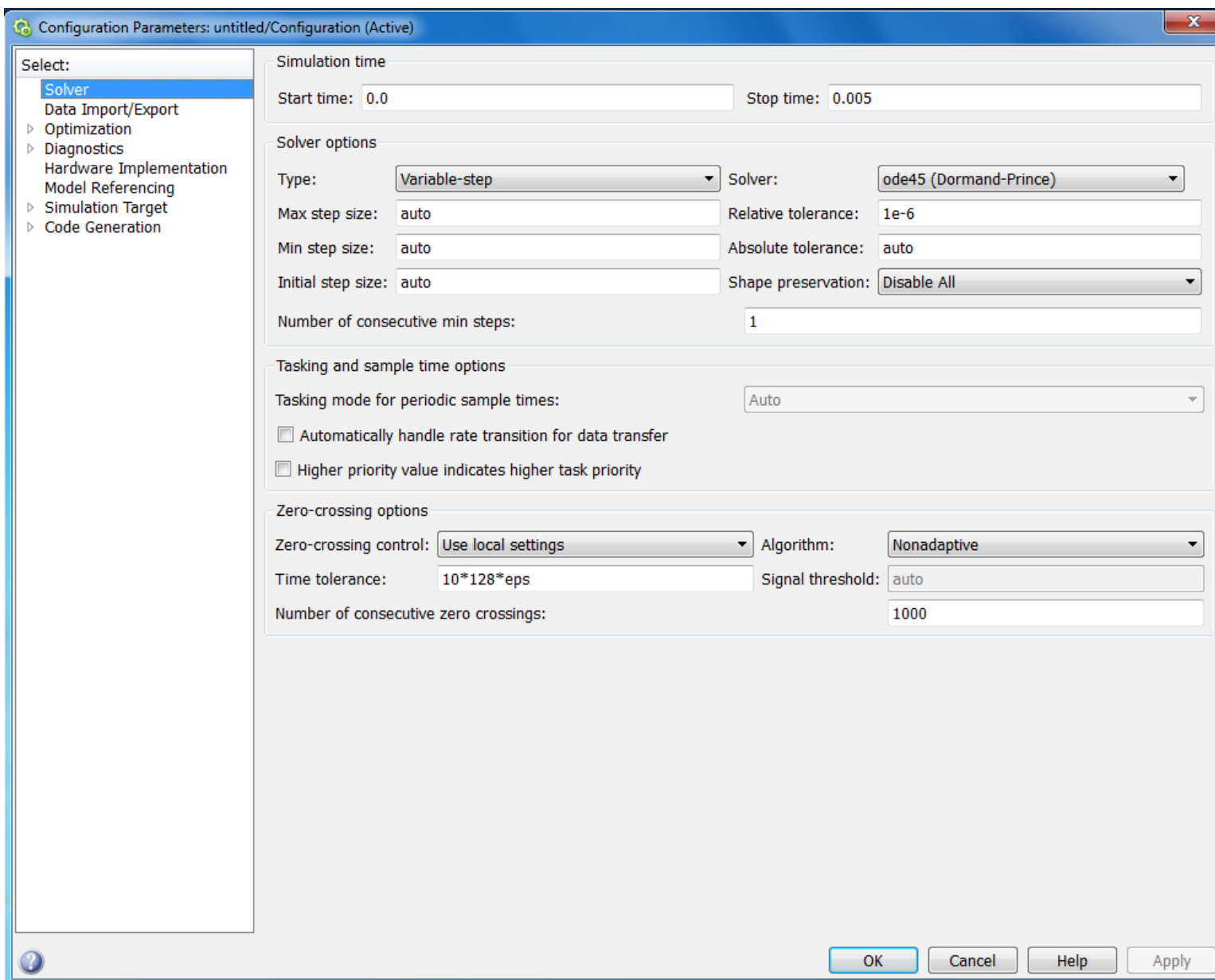


V okně lze provádět následující akce

- Z diferenciálních rovnic vytvářet bloková schémata,
- nastavovat parametry simulace (doba simulace, krok výpočtu, metoda výpočtu, atd.) a spouštět vlastní simulaci,
- nastavovat parametry a vlastnosti modelu,
- je možné spouštět různé analýzy (např. lineární analýza).

Základní nastavení parametrů simulace

- Nastavuje čas startu a zastavení simulace,
- velikost minimálního a maximálního kroku simulace,
- velikost tolerance,
- typ numerického výpočtu .



Configuration Parameters: untitled/Configuration (Active)

Select:

- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- Code Generation

Simulation time

Start time: 0.0 Stop time: 0.005

Solver options

Type: Variable-step Solver: ode45 (Dormand-Prince)

Max step size: auto Relative tolerance: 1e-6

Min step size: auto Absolute tolerance: auto

Initial step size: auto Shape preservation: Disable All

Number of consecutive min steps: 1

Tasking and sample time options

Tasking mode for periodic sample times: Auto

Automatically handle rate transition for data transfer

Higher priority value indicates higher task priority

Zero-crossing options

Zero-crossing control: Use local settings Algorithm: Nonadaptive

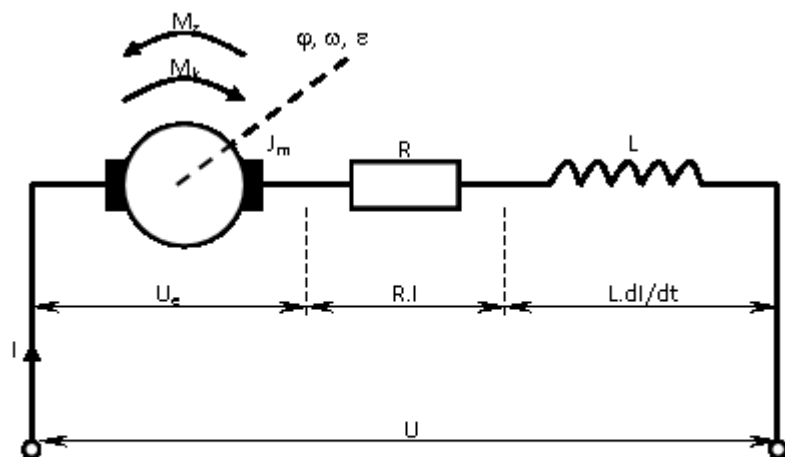
Time tolerance: 10*128*eps Signal threshold: auto

Number of consecutive zero crossings: 1000

OK Cancel Help Apply

Vytvoření blokového schématu stejnosměrného motoru

Nejdříve matematický popis stejnosměrného motoru



Elektrické schéma stejnosměrného motoru.

Matematické rovnice

$$U = U_e + R \cdot I + L \cdot \frac{dI}{dt}$$

$$M_k = K_{m1} \cdot I$$

$$U_e = K_{m2} \cdot \omega$$

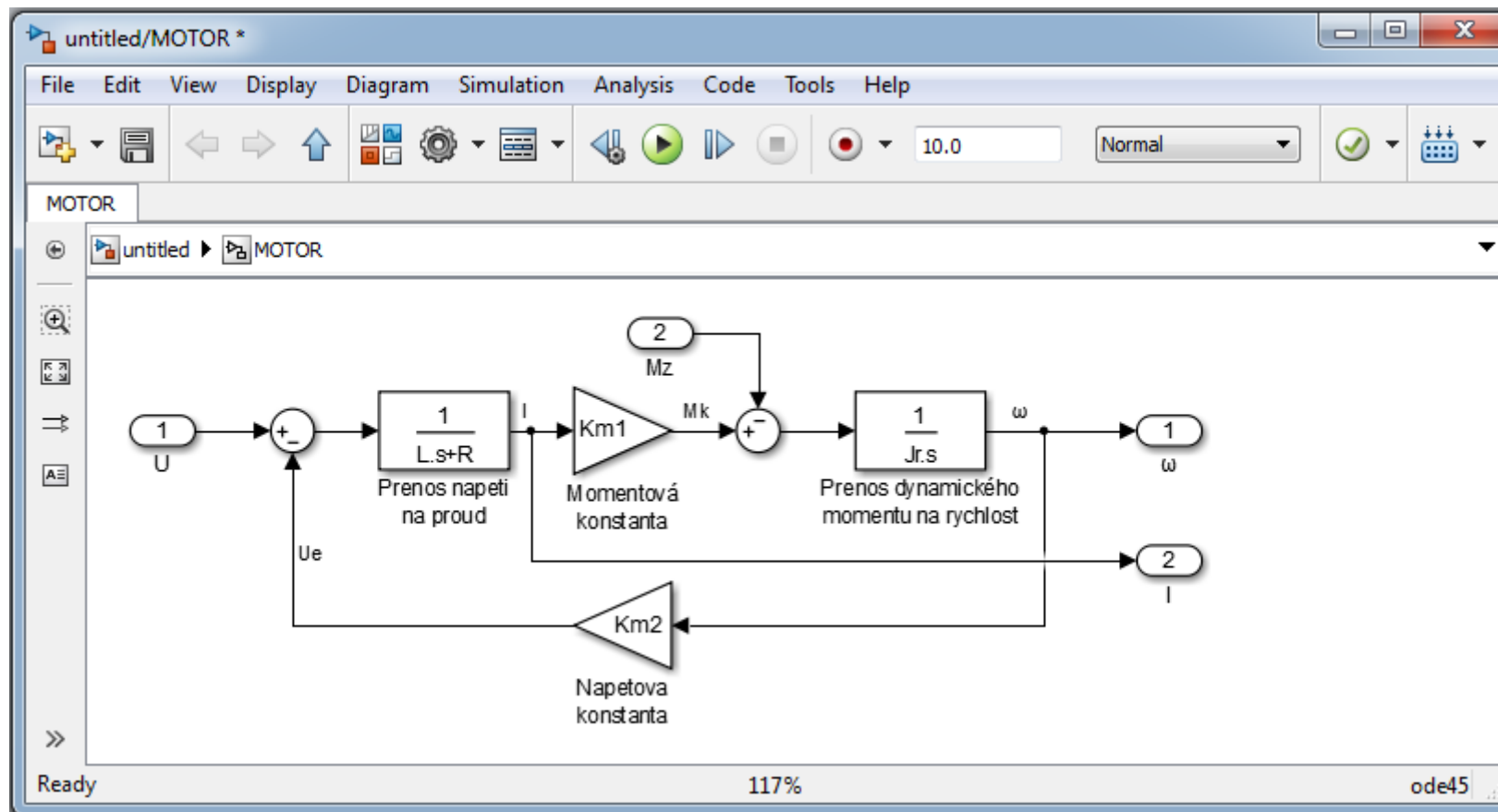
$$M_k - M_z = J_r \cdot \frac{d\omega}{dt}$$

Po dosazení a Laplaceově transformaci

$$U = U_e + R \cdot I + L \cdot I \cdot s \quad \rightarrow \quad I = (U - K_{m2}) \cdot \frac{1}{R + L \cdot s}$$

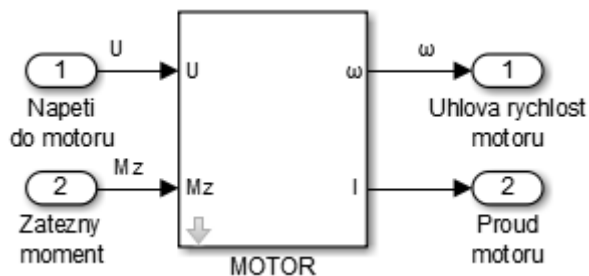
$$M_k - M_z = J_r \cdot \omega \cdot s \quad \rightarrow \quad \omega = (M_k - M_z) \cdot \frac{1}{J_r \cdot s}$$

Vytvoření blokového schématu stejnosměrného motoru

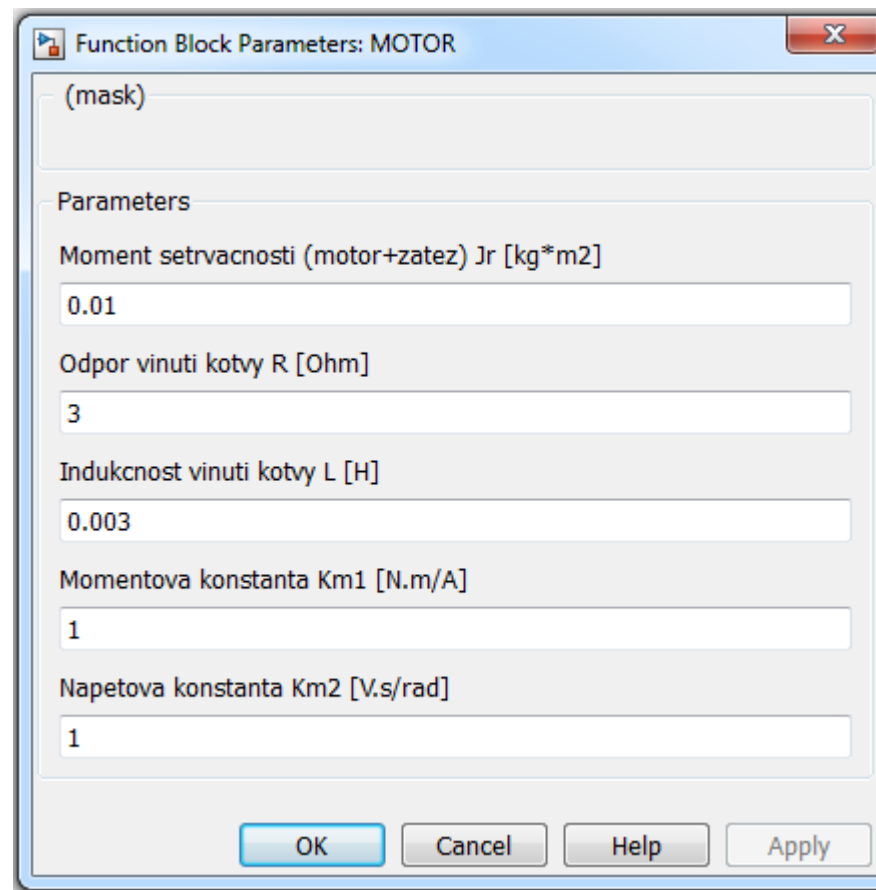


- Bloky jsou v knihovnách: *Continuous*, *Math Operation*, *Sources*, *Sinks*,
- nebo všechny bloky v jedné knihovně: *Commonly Used blocks*.

Vytvoření subsystému MOTOR

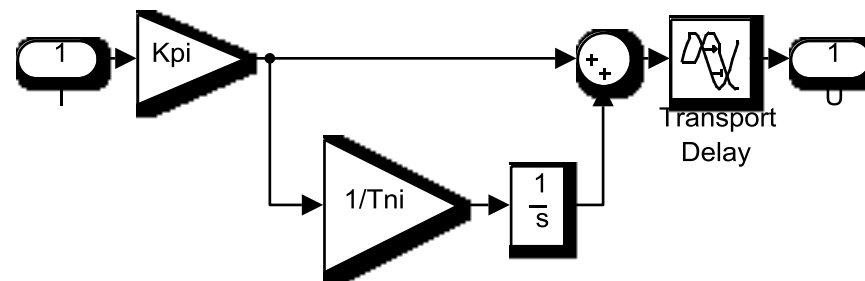


- Subsystém obsahuje 2 vstupy a 2 výstup a je vytvořený z knihovny *Ports & Subsystems*,
- dále je vytvořena maska subsystému, kde se zaznamenají hodnoty J_r , R , L , K_{m1} , K_{m2} .

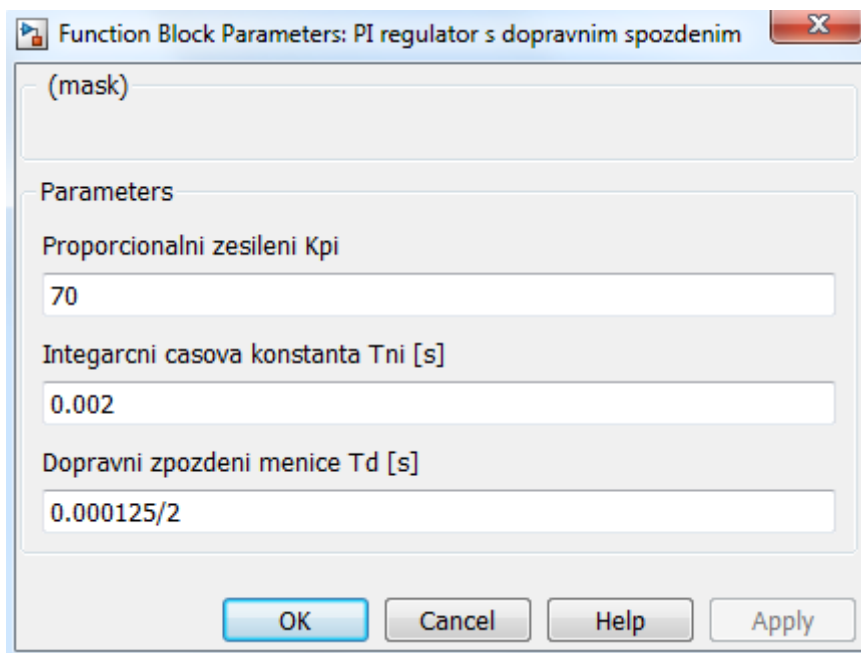
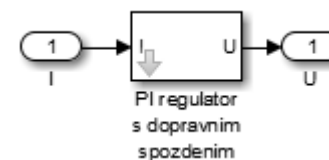


Proudový regulátor s měničem

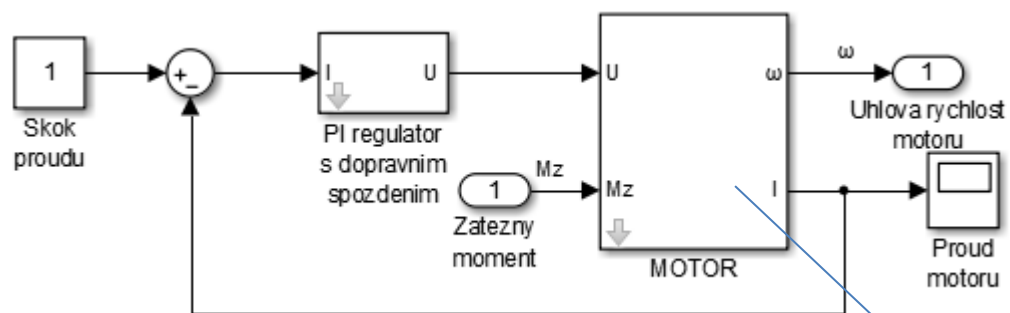
- Proudový regulátor je typu PI (K_{pi} , T_{ni}),
- měnič nahrazen dopravním zpožděním.



Subsystem proudového regulátoru

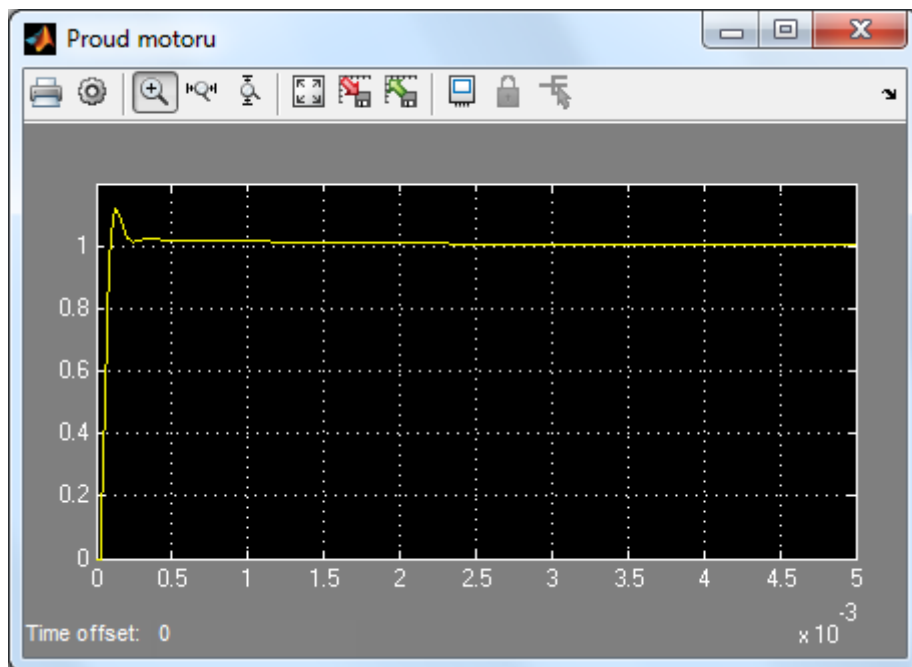


Proudová smyčka

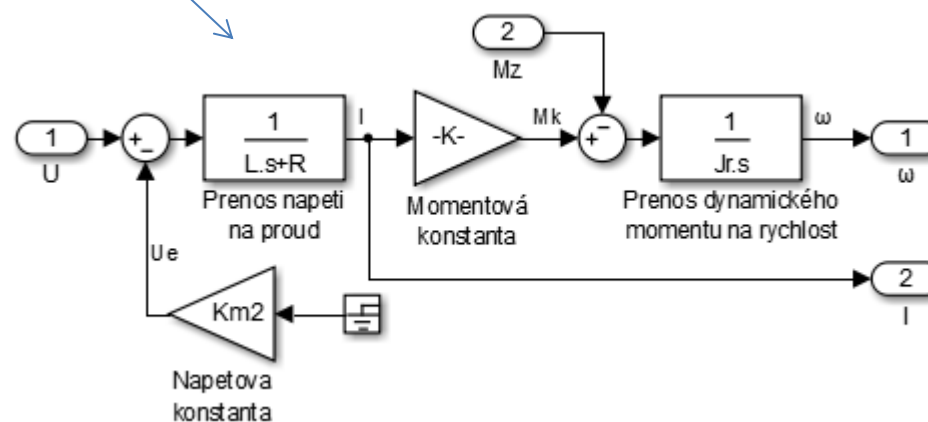


Měření i simulace se provádí při zablokované hřídeli motoru

Odezva na skok proudu

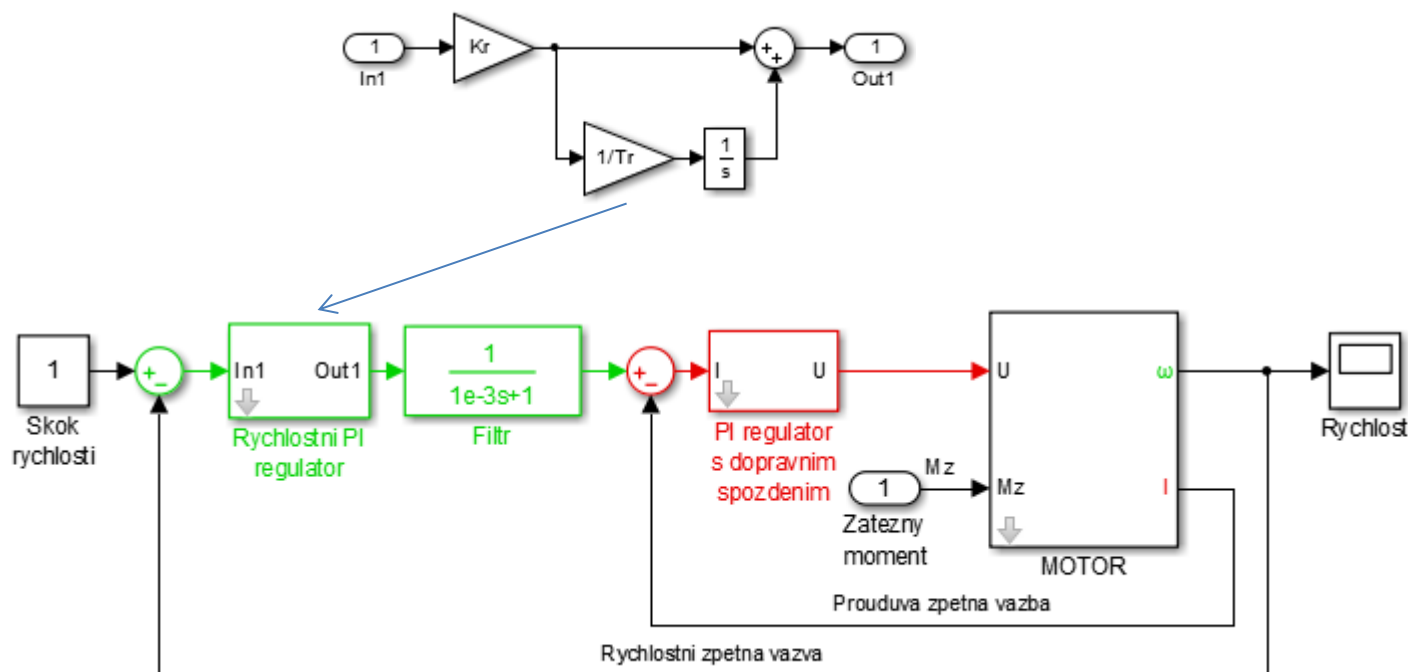


Upravený subsystém MOTOR pro měření proudové smyčky



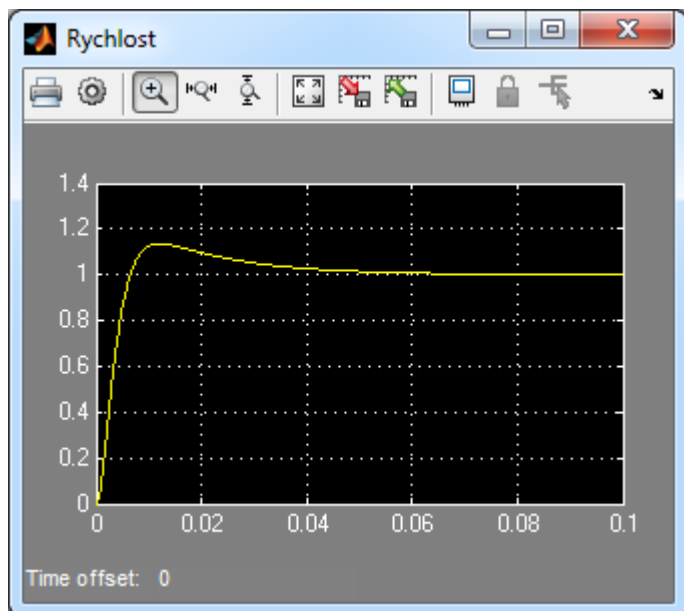
Rychlostní smyčka

- Rychlostní regulátor je typu PI (K_r , T_r)
- Filtr v rychlostní smyčce je pro odstranění nežádoucích vysokých frekvencí

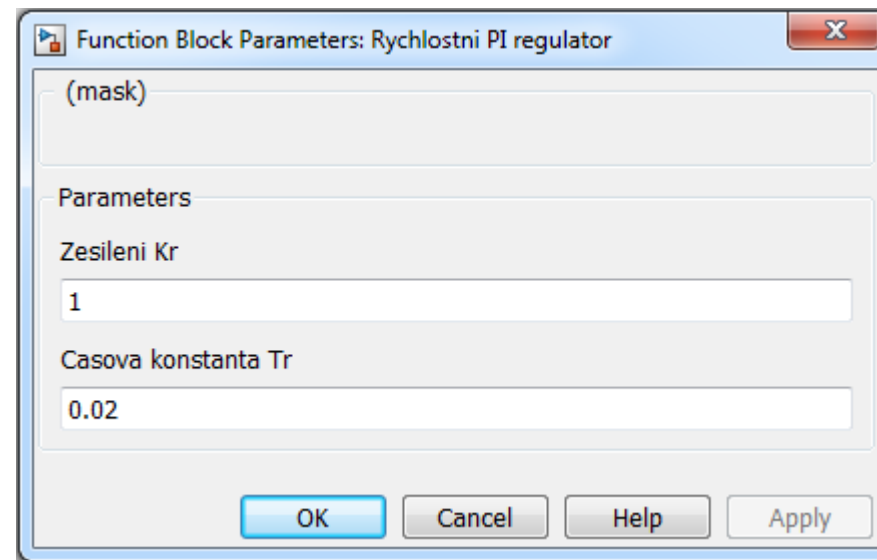


Rychlostní smyčka – nastavení regulátorů

Odezva na skok rychlosti



Parametry subsystému rychlostního regulátoru



Function Block Parameters: Rychlostni PI regulator

(mask)

Parameters

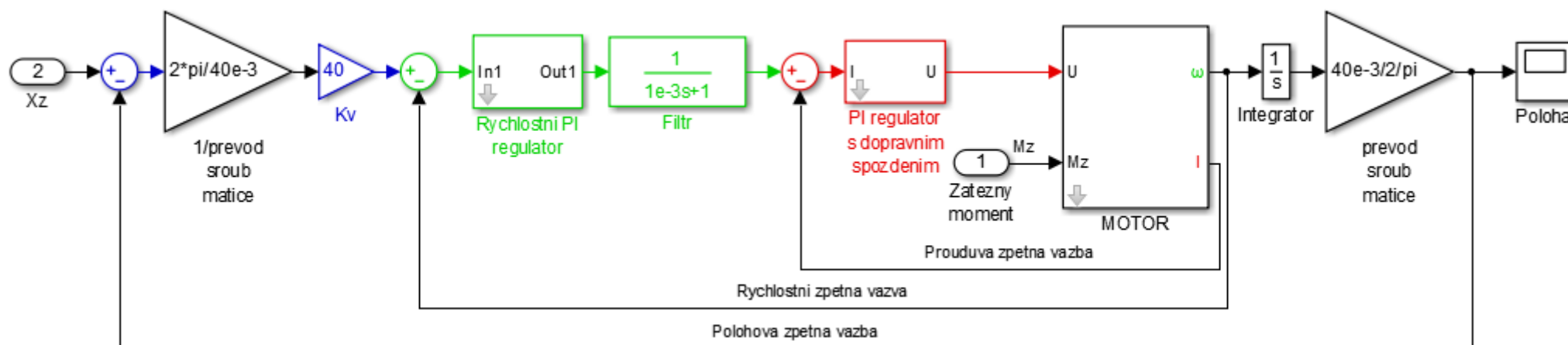
Zesileni Kr
1

Casova konstanta Tr
0.02

OK Cancel Help Apply

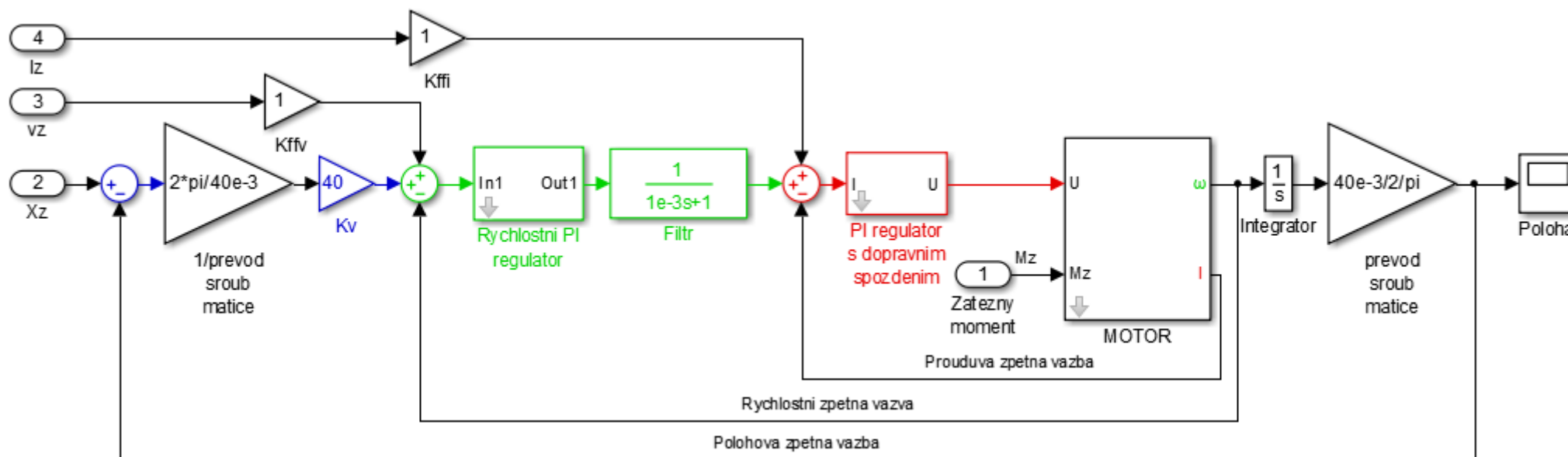
Polohová smyčka

- Polohový regulátor je typu P (zesílení K_v),
- kuličkový šroub má stoupání 40 mm/ot
- X_z – žádaná hodnota polohy (v metrech)



Zapojení feedforwardů

- Proudový feedforward (K_{ffi} – váha feedforwardu),
- rychlostní feedforward (K_{ffv} – váha feedforwardu),
- V_z – hodnota žádané rychlosti,
- I_z – hodnota žádaného proudu.



Děkuji za pozornost