

A3B35ARI – zadání úlohy LEGO Segway

Cílem této semestrální úlohy je řízení LEGO robota, v tomto případě ve tvaru dvoukolého vozítka SEGWAY (http://en.wikipedia.org/wiki/Segway_PT), jehož matematický model odpovídá inverznímu kyvadlu.

Pro zjednodušení práce budeme pracovat s toolboxem Lego Mindstorms NXT pro Matlab 2012b (a novější). Ty umožňují velmi efektivní návrh a ověření regulátoru v prostředí Simulink a následné nahrání řídicího algoritmu do Lego robota přes USB kabel nebo Bluetooth. Současně umožňuje on-line ladění konstant regulátoru bez nutnosti znovunahrání kódu a rovněž on-line vyčítání dat z robota přímo do Simulinku. Jedná se o novější verzi balíku, který původně vytvořil Yorihiya Yamamoto¹. Model robota (vozítka) včetně principu funkce simulinkového modelu je však totožný, proto budeme vycházet z jeho materiálu [NXTway-GS Model-Based Design.pdf](#), které si můžete stáhnout na Moodle.

Předpřipravené nástroje ovšem vyžadují používat OS Windows a mít nainstalovaný Matlab 2012b (nebo novější). Z technických důvodů bohužel není možné pracovat přímo na strojích v laboratořích K23 a K26. Proto si toolbox spolu s nutnými nástroji budete muset nainstalovat na svůj vlastní počítač. Tím také budete moci pracovat na úloze i mimo cvičení. Naštěstí je instalace všech potřebných programů velmi jednoduchá.

Aktualizace (25.3.2015):

- Verze Matlabu 2013 a výše již bohužel nepodporují online vyčítání hodnot přes USB. Nyní lze použít pouze Bluetooth. Nahrání kódu lze provést přes USB (rychlejší varianta) i Bluetooth.
- Verze Matlabu 2013 a výše vyžaduje vytvoření účtu na Mathworks stránkách. Za vytvoření účtu se neplatí žádný poplatek a ani není potřeba uvádět registrační číslo Matlabu. Stačí pouze email, jméno (klidně fiktivní), univerzita a heslo.

Cíl práce:

- 1) Sestavte robota podle níže uvedeného návodu.
- 2) Navrhněte stavový zpětnovazební regulátor, jenž bude stabilizovat jak úhel naklonění robota, tak jeho polohu (v podstatě všechny stavy).
- 3) Rozšiřte bod 2 o sledování reference na úhlovou rychlost koleček. Cílem je ujet za 15s co největší vzdálenost, poté zastavit a zůstat ve stabilizované svislé poloze.

Do stanoveného termínu nahrajte na Moodle podklady, ze kterých bude jasné, jak jste postupovali, jak jste navrhovali který regulátor a k čemu jste dospěli.

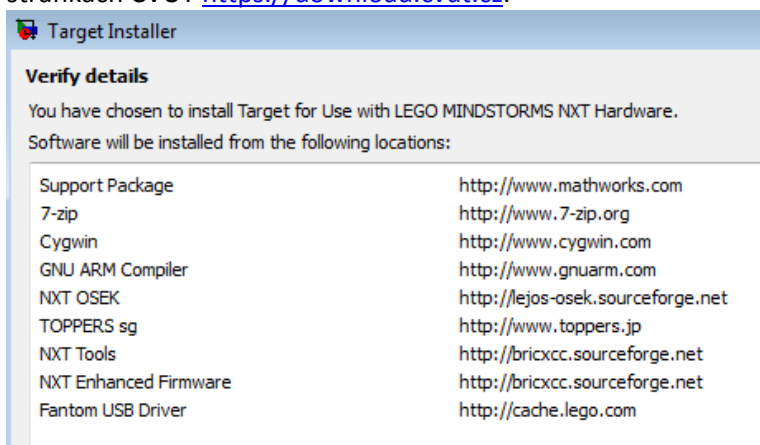
Sestavení robota

Detailní návod na sestavení robota je v souboru [NXTway-GS Building Instructions.pdf](#). V tomto bodě je volitelná poloha a spojení kol – je možnost mít široký/úzký rozchod kol a také (ne)spojenou jejich nápravu. Design robota či různé doplňky můžete samozřejmě upravit podle týmové preference :-), pozor však na to, že větší změnou můžete změnit dynamické vlastnosti robota a nemusí pak odpovídat model.

¹ <http://www.mathworks.com/matlabcentral/fileexchange/19147-nxtway-gs-self-balancing-two-wheeled-robot-controller-design>

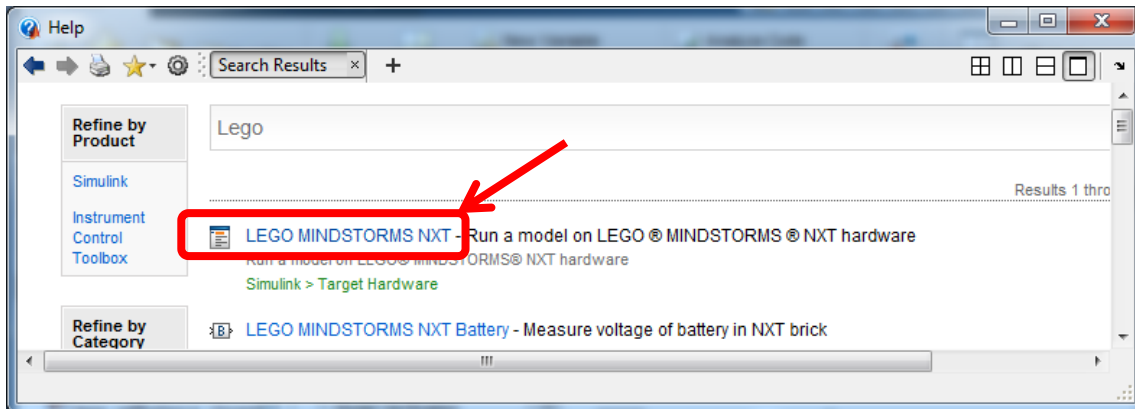
Instalace Lego toolboxu do Matlabu

Pro Lego Mindstorms existuje toolbox do Matlabu, který je možné si zdarma stáhnout a nainstalovat, bohužel je kompatibilní pouze s verzí 2012b a novější. Současně se nainstalují i další programy, které Lego toolbox pro svou činnost potřebuje (viz Obr.1). Nejnovější verze Matlabu je ke stažení na stránkách ČVUT <https://download.cvut.cz>.

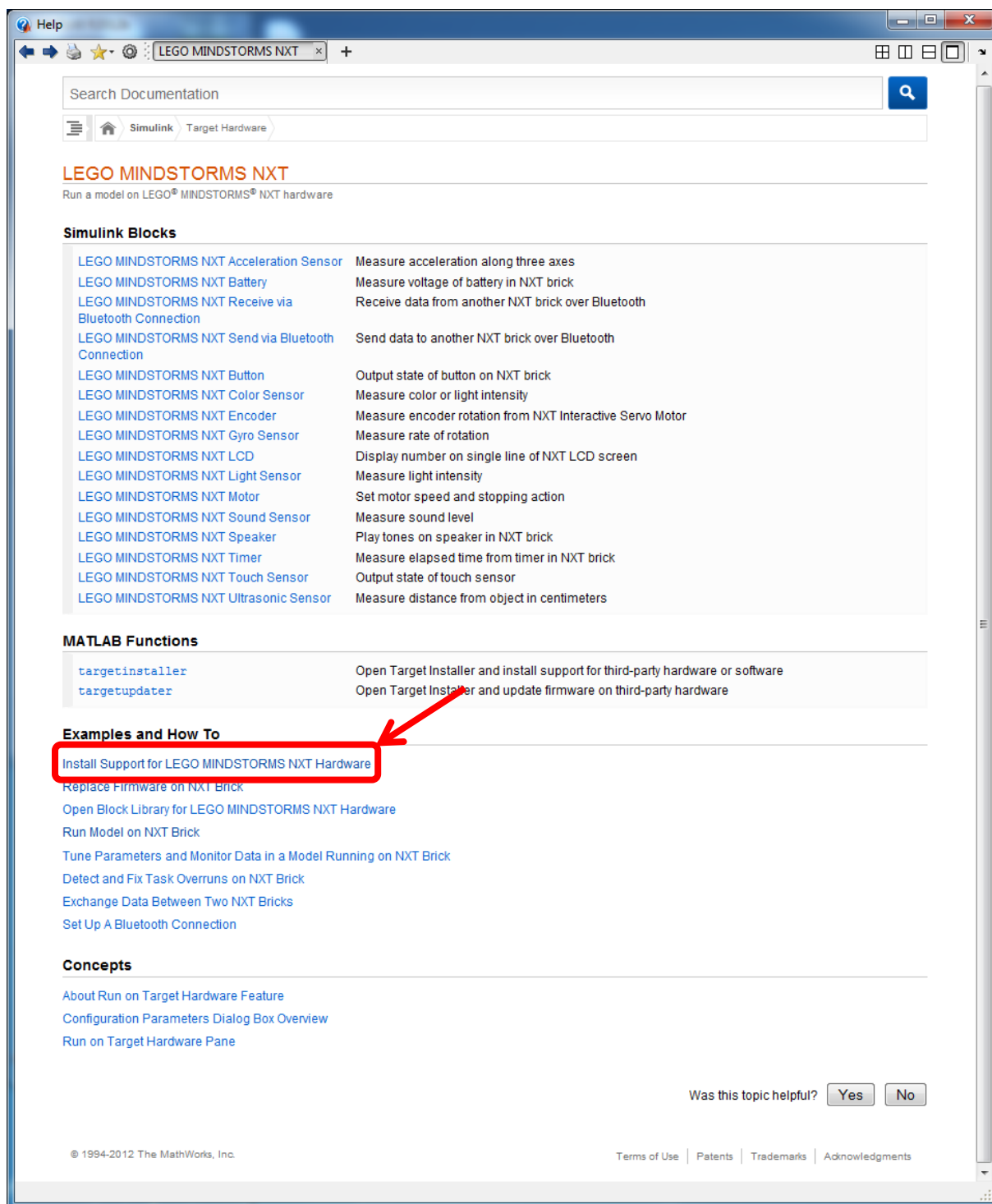


Obr. 1 Seznam programů, které LEGO toolbox potřebuje pro svou funkci.

Podrobný návod instalace je k dispozici na stránkách Mathworks <http://www.mathworks.com/hardware-support/lego-mindstorms-simulink.html> nebo přímo v nápovědě Matlabu. Stačí otevřít Help a vyhledat slovo Lego (viz. Obr.2) (**Simulink -> Target Hardware -> LEGO MINDSTORMS NXT**)

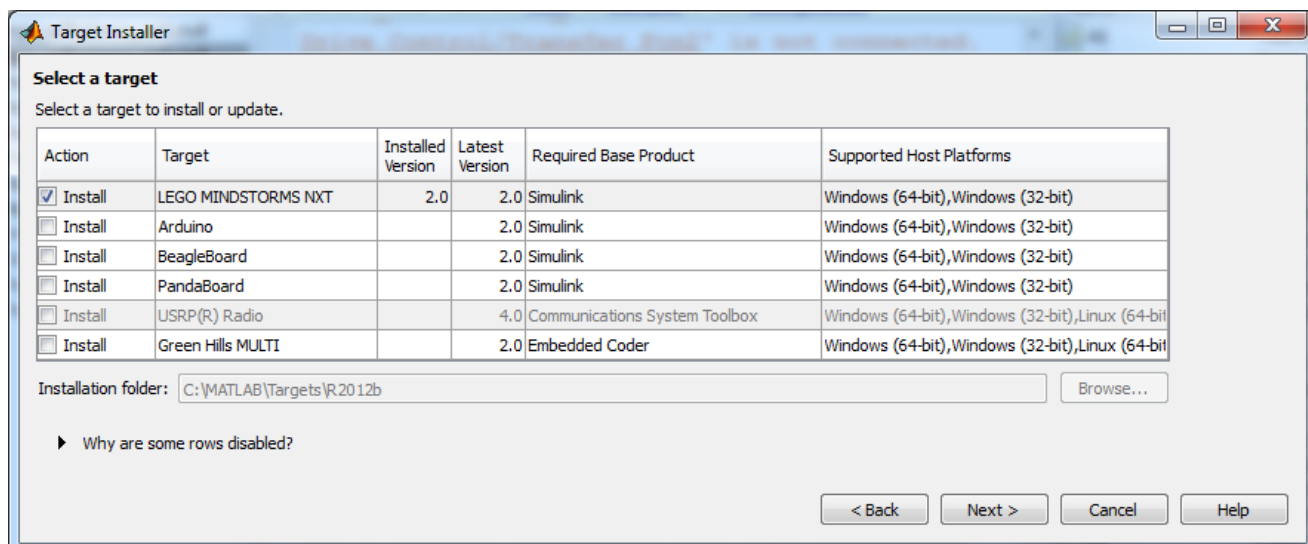


Obr. 2 Vyhledání klíčového slova "Lego" v Helpu Matlabu



Obr. 3 Matlab 2012 už obsahuje nápovědu pro LEGO MINDSTORMS

Samotná instalace toolboxu se spustí příkazem *targetinstaller*. Vyberete, že chcete nainstalovat Lego Mindstorms NXT a pak už postupujete podle pokynů instalátoru.



Obr. 4 Nabídka po zavození příkazu *targetinstaller* (zaškrtněte LEGO MINDSTORMS NXT)

Simulinkový model (*lego_selfbalance_students.slx*)

Simulinkové schéma *lego_selfbalance_students.slx* je rozděleno do dvou částí, do částí Hardware a Controller.

Ke spuštění modelu *lego_selfbalance_students.slx* je potřeba mít ve stejném adresáři mfile *lego_selfbalance_parameters.m*, který obsahuje všechny konstanty pro simulinkový model.

Hardware

Blok Hardware má dvě funkce.

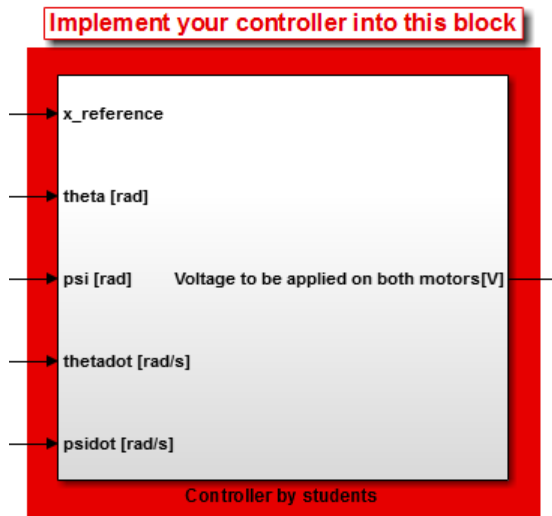
- Obsahuje všechny bločky, které zajišťují správný chod kódu po nahrání do robota (posílání střídů na motory, vyčítání dat ze senzorů).
- Je zde matematický model Lego robota, který umožňuje vyzkoušet si navržené regulátory ještě před tím, než budou nahrány do robota.

Controller

V části Controller je řídicí program pro Lego robota. Skládá se z následujících bločků:

- Blok *Controller/Gyro Calibration* kalibruje gyro senzor robota. Kalibrace probíhá první čtyři vteřiny běhu programu. Poznává se podle toho, že robot pípá. **Během kalibrace je nutné robota držet ve vertikální poloze.** Kalibrace neprobíhá, pokud program běží v Simulinku na počítači.
- Blok *Controller/Balance and Drive Control/Signals scaling and enhancing* přepočte a naškáluje údaje změřené senzory robota. Výstupem jsou změřené (a dopočítané) stavy $[\theta, \psi, \dot{\theta}, \dot{\psi}]$.
- Blok *Controller/Balance and Drive Control/Reference states generator* generuje reference na stavy podle zadané rychlosti robota $speed_{ref}$. To se Vám později bude hodit. Možná si tento bloček budete chtít upravit v závislosti na Vašem regulátoru.

- Do bloku **Controller/Balance and Drive Control/Controller by students** **naimplementujte Váš regulátor**. Vstupem jsou čtyři změřené stavy $[\theta, \psi, \dot{\theta}, \dot{\psi}]$ a vektor referencí na tyto stavy. Výstupem je napětí, které se aplikuje na motory.



Obr. 5a Blok kam naimplementujete regulátor



Obr. 5b Blok generující reference na stavy

- Blok **Controller/Balance and Drive Control/PWM scaling of DC motors and wheels synchronization** škáluje napětí vypočtené regulátorem podle aktuálního stavu baterie a přepočítá ho na velikost střidy PWM, která se aplikuje na motory. Současně se vypočítá a aplikuje korekce, která zajistí, aby se obě kolečka motoru otočila o stejný úhel θ .

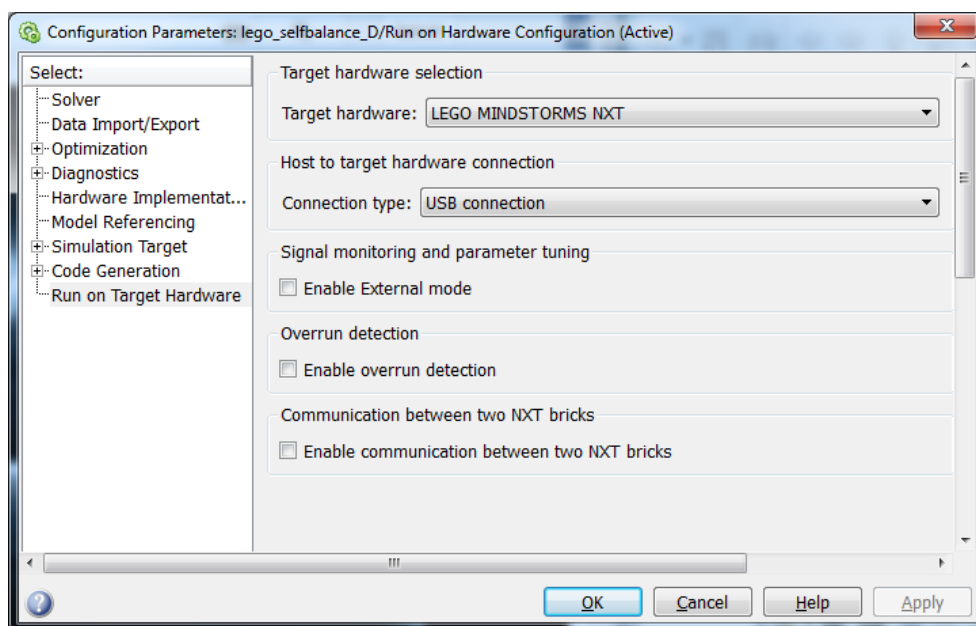
Nahrání kódu do robota

Aktualizace (25.3.2015):

- V novějších verzích Matlabu již není **Run** v nabídce „Run on target hardware“. Program se spustí přepnutím do módu **External** a standartním příkazem **Run** (zelená šipka). V módu **Normal** pak běží simulace s matematickým modelem robota.
- Pokud chcete pouze nahrát nový kód do robota a nechcete vyčítat data ze senzorů, tak stačí zavolat příkaz **Deploy to Hardware** (CTRL+B).

Detailní návod, jak nahrát kód do robota najdete v Helpu po zadání „Lego Mindstorms Run model on NXT Brick“ nebo na stránkách Mathworks (odkaz uveden výše). Přesto si ho tu raději ve zkratce popíšeme:

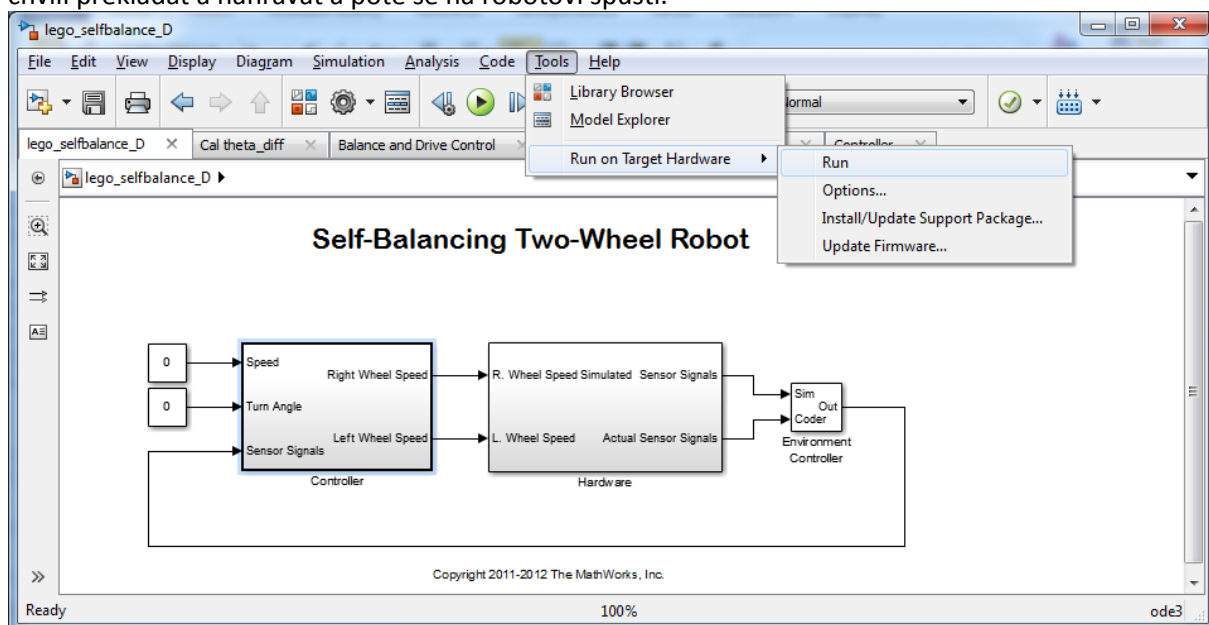
- 1) Připojte pomocí USB kabelu robota a zapněte ho. V simulinkovém modelu robota otevřete **Tools -> Run on Target Hardware -> Update firmware**. Tím do robota nahrajete potřebný firmware.
- 2) Dále otevřete **Tools -> Run on Target Hardware -> Options -> Run On Target Hardware** a ujistěte se, že je nastaveno *Target hardware: LEGO MINDSTORMS NXT* a *Connection type: USB connection*. Je možné nastavit i Bluetooth připojení, ale obvykle je potřeba manuálně nastavit číslo COM portu, na kterém máte robota zaregistrovaného (viz. Help).



Obr. 6 Nastavení komunikace s LEGO MINDSTORMS NXT

Kroky 1) a 2) stačí **vykonat pouze jednou**. Jakmile je jednou provedete, tak firmware i nastavení komunikace zůstane uloženo.

- 3) Kód nahrajete a spustíte kliknutím na **Tools -> Run on Target Hardware -> Run**. Kód se bude chvíli překládat a nahrávat a poté se na robotovi spustí.



Obr. 7 Ukázka nahrání a spuštění kódu

- 4) Pokud jste kód nezměnili a chcete ho jen spustit, tak je uložen v paměti robota v **My Files -> Software Files -> Váš program**

Model robota

Matematický model robota je popsán v [NXTway-GS Model-Based Design.pdf](#) (na str.6 je graficky znázorněn význam jednotlivých stavů). Linearizovaný model, se kterým pracuje simulinkové schéma je následující

$$\dot{x}_1 = A_1 x_1 + B_1 u \quad A_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -410 & -162 & 162 \\ 0 & 270 & 78 & -78 \end{pmatrix} \quad B_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 158 & 158 \\ -76 & -76 \end{pmatrix},$$

kde vektor stavů $x_1 = [\theta, \psi, \dot{\theta}, \dot{\psi}]^T$, $u = [v_l, v_r]^T$

θ – úhel natočení koleček ψ – úhel náklonu robota v_l – napětí na levém motoru

$\dot{\theta}$ – úhlová rychlost koleček $\dot{\psi}$ – úhl. rychl. náklonu robota v_r – napětí na pravém motoru

Jako pracovní bod je brána vertikální poloha robota, tedy $\phi = 0$ [rad], $\psi = 0$ [rad], $\dot{\phi} = 0$ [rad/s], $\dot{\psi} = 0$ [rad/s].

Rady na závěr

- Dejte si pozor na saturaci akčního zásahu. Hodnota Voltage, která je výstupem regulátoru, je napětí, které chcete aplikovat na motory. Jelikož nemáte k dispozici tvrdý zdroj napětí ale baterii, tak se toto napětí přeskákuje v závislosti na stavu napětí (=nabití) baterie a dále se přepočítá na střidu PWM. Počítejte s tím, že baterie dodá přibližně 8V. Tato hodnota je současně použita i v simulinkovém modelu, proto navrhnete takový regulátor, který na výstupu nebude mít hodnotu příliš větší než 8V (pozn. maximální hodnota střidy je 100%).
- Při implementaci regulátoru je nutno dbát na správné datové typy, které lze v případě potřeby konvertovat pomocí bloku *Data type conversion*.
- Dejte si pozor na to, aby regulátor byl diskretní v čase, lze například použít blok *Discrete Transfer Fcn* nebo bloček *PID Controller* a nastavit periodu vzorkování.