# Benchmark: Reachability on a model with holes

Thomas Heinz[*], Jens Oehlerking[*], Matthias Woehrle[*]

March 28, 2014

**Abstract**

The benchmark presented in this paper is an example for verification of a hybrid system model with so-called holes, i.e. part of the system behaviour is not specified. Verification of such a model allows 3$^{\text{rd}}$ parties to plug a specific behaviour into a hole without changing desirable properties of the system. The particular example is based on an open-source robotics application, namely a self-balancing two-wheeled robot which is essentially modeled as an inverted pendulum. The balance controller provides two input signals for translational (forward/backward) and rotational (left/right turn) motion which can be driven by arbitrary path planning applications. Examples for such applications are line following and pursuit-evasion algorithms as well as a remote control which allows trajectories to be defined externally. These motion trajectories may or may not yield a state from which the balance controller is unable to recover, which means that the robot falls over. Hence, the verification goal is to prove the safety property that the body pitch angle is bounded under motion trajectories.

**Category:** industrial **Difficulty:** medium

## 1 Context and Origins

Figure 1 depicts an abstract model of a system which consists of a controller and a plant in a feedback loop. The inputs of the system are controller or plant inputs. A controller input can often be characterised as a command in the sense that it modifies the goal of the controller dynamically. A plant input is often used to model uncertainties by so-called disturbances. A hole in the context of this paper is an unknown function that defines the system input. A common practical use case is a cascaded control design where from the point of view of the inner controller, the outer one can be considered as a hole. The verification goal is to prove relevant properties of the inner controller regardless of the definition of the outer one.

[*]Robert Bosch GmbH, Germany

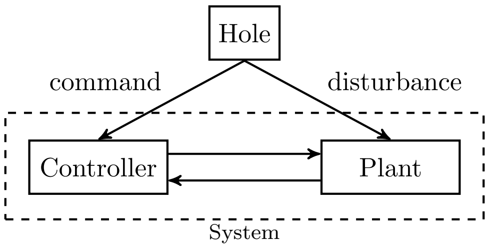
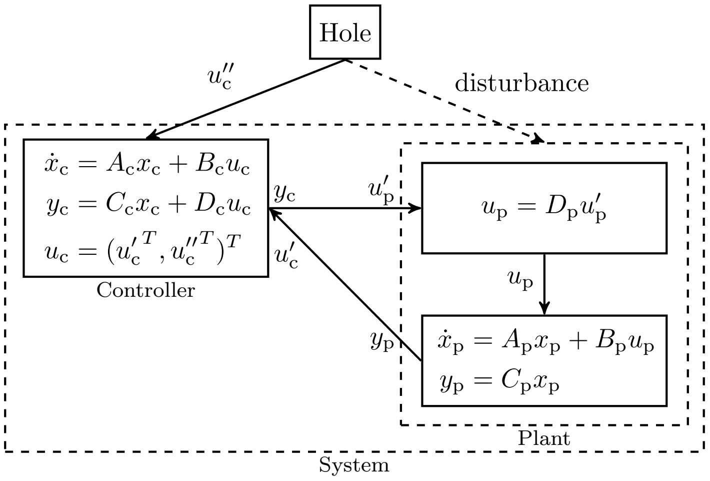


Figure 1: System (controller and plant) model with a hole.



Hole
$u''_\mathrm{c}$
disturbance
$\dot{x}_\mathrm{c} = A_\mathrm{c} x_\mathrm{c} + B_\mathrm{c} u_\mathrm{c}$
$y_\mathrm{c} = C_\mathrm{c} x_\mathrm{c} + D_\mathrm{c} u_\mathrm{c}$
$u_\mathrm{c} = ({u'_\mathrm{c}}^T, {u''_\mathrm{c}}^T)^T$
Controller
$y_\mathrm{c}$
$u'_\mathrm{p}$
$u'_\mathrm{c}$
$u_\mathrm{p} = D_\mathrm{p} u'_\mathrm{p}$
$u_\mathrm{p}$
$y_\mathrm{p}$
$\dot{x}_\mathrm{p} = A_\mathrm{p} x_\mathrm{p} + B_\mathrm{p} u_\mathrm{p}$
$y_\mathrm{p} = C_\mathrm{p} x_\mathrm{p}$
Plant
System

Figure 2: NXTway-GS as a model with a hole.

## 2 Brief description

The particular case study for such a model with holes described below is a self-balancing two-wheeled robot called NXTway-GS[1] by Yorihisa Yamamoto. The robot is constructed from the LEGO Mindstorms$^{\mathrm{TM}}$NXT 2.0 kit and modeled as an inverted pendulum. The NXTway-GS package bundles building instructions, a Matlab/Simulink model composed of a continuous plant and discrete time controller model and a detailed description of the model. To avoid redundancy, the model is only briefly described below. For further details, the reader is referred to the NXTway-GS documentation. Figure 2 provides an overview of the system as well as the notation used in the following.

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

## 2.1 Plant model

The plant is a two-wheeled robot where both wheels are driven by independently controlled servo motors. The plant as depicted on the right of Figure 2 relates motor voltage of the left and right servo motor $v_\mathrm{l}, v_\mathrm{r}$ to the left and right motor angle $\theta_{\mathrm{m}_\mathrm{l}}, \theta_{\mathrm{m}_\mathrm{r}}$ and the body pitch angular velocity $\dot{\psi}$. The state vector $x_\mathrm{p} = (\dot{\theta}, \theta, \dot{\psi}, \psi, \dot{\phi}, \phi)^T$ is composed of the average angle of left and right wheel $\theta$, body pitch angle $\psi$, body yaw angle $\phi$, and the respective angular velocities. The actual input of the plant is an encoding of $v_\mathrm{l}, v_\mathrm{r}$ as pulse-width modulation duty cycles $p_\mathrm{l}$, $p_\mathrm{r}$. A preprocessing block relates $u'_\mathrm{p} = (p_\mathrm{l}, p_\mathrm{r})^T$ to $u_\mathrm{p} = (v_\mathrm{l}, v_\mathrm{r})^T$ by $u_\mathrm{p} = D_\mathrm{p} u'_\mathrm{p}$ based on the following equations where $V_B$ is the battery voltage which is assumed to be constant.

$$\begin{aligned} v_\mathrm{l} &= \frac{p_\mathrm{l}}{100}(0.001089 \cdot V_B - 0.625) \\ v_\mathrm{r} &= \frac{p_\mathrm{r}}{100}(0.001089 \cdot V_B - 0.625) \end{aligned}$$

The output of the plant is $y_\mathrm{p} = (\dot{\psi}_\mathrm{out}, \theta_{\mathrm{m}_\mathrm{l}}, \theta_{\mathrm{m}_\mathrm{r}})^T$ given by $y_\mathrm{p} = C_\mathrm{p} x_\mathrm{p}$ based on the following equations where $W$ is the body width and $R$ is the wheel radius.

$$\begin{aligned} \dot{\psi}_\mathrm{out} &= \dot{\psi} \\ \theta_{\mathrm{m}_\mathrm{l}} &= \frac{180}{\pi}\left(\theta - \psi - \frac{\phi W}{2R}\right) \\ \theta_{\mathrm{m}_\mathrm{r}} &= \frac{180}{\pi}\left(\theta - \psi + \frac{\phi W}{2R}\right) \end{aligned}$$

The state space equation of the plant $\dot{x}_\mathrm{p} = A_\mathrm{p} x_\mathrm{p} + B_\mathrm{p} u_\mathrm{p}$ is derived from a nonlinear model of the inverted pendulum motion equations as well as the DC motor equations. Both systems of equations define the angular forces $F_\theta$, $F_\psi$, $F_\phi$ as follows where $F_\star = f_i(\ldots)$ represent the motion equations and $F_\star = g_i(\ldots)$ are the DC motor equations. The definition of the $f_i$, $g_i$ can be found in the NXTway-GS documentation.

$$\begin{aligned} f_1(\ddot{\theta}, \ddot{\psi}, \dot{\psi}, \psi) &= F_\theta = g_1(\dot{\theta}, \dot{\psi}, v_\mathrm{l}, v_\mathrm{r}) \\ f_2(\ddot{\theta}, \ddot{\psi}, \psi, \dot{\phi}) &= F_\psi = g_2(\dot{\theta}, \dot{\psi}, v_\mathrm{l}, v_\mathrm{r}) \\ f_3(\dot{\psi}, \psi, \ddot{\phi}, \dot{\phi}) &= F_\phi = g_3(\dot{\phi}, v_\mathrm{l}, v_\mathrm{r}) \end{aligned}$$

In the next step, the $f_i$ are linearised around the balance point $\psi = 0$ yielding a linear system of equations as the $g_i$ are already linear. The system can be easily transformed into the standard state space form. The matrices $A_\mathrm{p}$, $B_\mathrm{p}$, $C_\mathrm{p}$, $D_\mathrm{p}$ derived using the constants described in the NXTway-GS documentation can be found in the appendix.

Note that the given plant model is deterministic. In order to account for inaccuracies, it may be generalised to a nondeterministic model reflecting disturbances as additive uncertainty, i.e. $\dot{x}_\mathrm{p} = A_\mathrm{p} x_\mathrm{p} + B_\mathrm{p} u_\mathrm{p} + E_\mathrm{p} d_\mathrm{p}$, $d_\mathrm{p} \in \mathbf{d}_\mathrm{p}^n \subseteq \mathbb{R}^n$, or multiplicative uncertainty, i.e. $\dot{x}_\mathrm{p} = A_\mathrm{p} x_\mathrm{p} + B_\mathrm{p} u_\mathrm{p}$, $A_\mathrm{p} \in \mathbf{A}_\mathrm{p}^{n \times n} \subseteq \mathbb{R}^{n \times n}$, $B_\mathrm{p} \in \mathbf{B}_\mathrm{p}^{n \times m} \subseteq \mathbb{R}^{n \times m}$, or both where $n = \dim(x_\mathrm{p})$, $m = \dim(u_\mathrm{p})$. The sets $\mathbf{d}_\mathrm{p}^n$, $\mathbf{A}_\mathrm{p}^{n \times n}$, $\mathbf{B}_\mathrm{p}^{n \times m}$ act as bounds on the uncertainty.

## 2.2 Controller model

The balance controller model on the left of Figure 2 is derived in two steps. First, a servo controller is derived from the linearised plant model by the linear quadratic regulator method. The controller maintains balance under translational (forward/backward) motion, i.e. its control goal is to steer the translational speed towards the reference value $\dot{\theta}_{\mathrm{ref}}$ and to steer the body pitch angle $\psi$ towards 0. The output of the servo controller is a motor voltage $v$ which yields translational motion when applied to both servo motors ($v = v_{\mathrm{l}} = v_{\mathrm{r}}$). In a second step, rotational motion is introduced by a rotational speed reference value $\dot{\phi}_{\mathrm{ref}}$. Rotational motion is achieved by adding $k\dot{\phi}_{\mathrm{ref}}$ to $v_{\mathrm{l}}$ and subtracting it from $v_{\mathrm{r}}$ for a suitable constant $k$. Both reference values form the system input $u''_{\mathrm{c}} = (\dot{\theta}_{\mathrm{ref}}, \dot{\phi}_{\mathrm{ref}})^T$. The controller input is $u_{\mathrm{c}} = ({u'_{\mathrm{c}}}^T, {u''_{\mathrm{c}}}^T)^T$ where $u'_{\mathrm{c}} = y_{\mathrm{p}}$. The output of the controller is the input of the plant, i.e. $y_{\mathrm{c}} = u'_{\mathrm{p}}$. The state vector is $x_{\mathrm{c}} = (\theta_{\mathrm{err}}, \theta_{\mathrm{ref}}, \dot{\theta}_{\mathrm{ref\_lpf}}, \psi, \theta_{\mathrm{lpf}})$. The state variables are informally described below.

- $\theta_{\mathrm{err}}$: Integration of error between $\dot{\theta}_{\mathrm{ref}}$ and actual $\dot{\theta}$
- $\theta_{\mathrm{ref}}$: Integration of $\dot{\theta}_{\mathrm{ref}}$
- $\dot{\theta}_{\mathrm{ref\_lpf}}$: Low pass filter applied to $\dot{\theta}_{\mathrm{ref}}$
- $\psi$: Body pitch angle (by integration of $\dot{\psi}$)
- $\theta_{\mathrm{lpf}}$: Low pass filter applied to the average value of left/right motor angle

The continuous time controller is governed by the equations $\dot{x}_{\mathrm{c}} = A_{\mathrm{c}}x_{\mathrm{c}} + B_{\mathrm{c}}u_{\mathrm{c}}$, $y_{\mathrm{c}} = C_{\mathrm{c}}x_{\mathrm{c}} + D_{\mathrm{c}}u_{\mathrm{c}}$. It is obtained by transforming the difference equations of the discrete time NXTway-GS controller model into differential equations. The actual difference equations are extracted from the C code that is generated from the Matlab/Simulink model[2]. The discrete time controller from the implementation is given by the equations $x_{\mathrm{c}}[k+1] = \hat{A}_{\mathrm{c}}x_{\mathrm{c}}[k] + \hat{B}_{\mathrm{c}}u_{\mathrm{c}}[k]$, $y_{\mathrm{c}}[k] = C_{\mathrm{c}}x_{\mathrm{c}}[k] + D_{\mathrm{c}}u_{\mathrm{c}}[k]$. The sampling rate is 4 ms. All matrices can be found in the appendix.

Note that the linear controller model is not realistic as it assumes unbounded actuators. Bounded actuators can be reflected in the model by saturation. In the concrete example, $p_{\mathrm{l}}$, $p_{\mathrm{r}}$ are saturated, i.e. $p_{\mathrm{l}}, p_{\mathrm{r}} \in [-100, 100]$. Figure 3 depicts a simple representation of saturation as a hybrid automaton.

## 2.3 Model of the hole

A hole in the case study represents an arbitrary function defining $u''_{\mathrm{c}}$ (cf. Figure 2). A direct way to encode this into the controller model is via additive uncertainty. Let $B'_{\mathrm{p}}$ be the matrix of the first $\dim(u'_{\mathrm{c}})$ column vectors of $B_{\mathrm{p}}$ and $B''_{\mathrm{p}}$ be the matrix of the last $\dim(u''_{\mathrm{c}})$ column vectors of $B_{\mathrm{p}}$. The controller equations with additive uncertainty are $\dot{x}_{\mathrm{c}} = A_{\mathrm{c}}x_{\mathrm{c}} + B'_{\mathrm{c}}u'_{\mathrm{c}} + B''_{\mathrm{c}}u''_{\mathrm{c}}$, $y_{\mathrm{c}} = C_{\mathrm{c}}x_{\mathrm{c}} + D'_{\mathrm{c}}u'_{\mathrm{c}} + D''_{\mathrm{c}}u''_{\mathrm{c}}$ with $u''_{\mathrm{c}} \in [-100, 100] \times [-100, 100]$. If additive uncertainty cannot be expressed

[2]The C code, from which the continuous controller model is derived, can be found in version 2.18 of the nxtOSEK project available at `http://sourceforge.net/projects/lejos-osek/files/nxtOSEK/nxtOSEK_v218.zip/download`. The code is located in the file `nxtOSEK/ecrobot/nxtway_gs_balancer/balancer.c` in the archive.

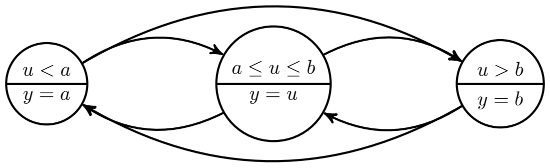
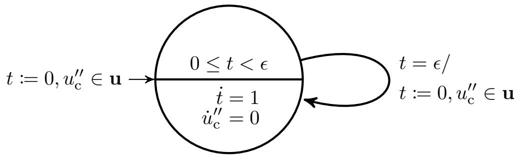


Figure 3: Saturation with input $u$ and output $y = \mathrm{sat}(u, [a, b])$.



$t := 0, u''_\mathrm{c} \in \mathbf{u}$
$0 \leq t < \epsilon$
$\dot{t} = 1$
$\dot{u}''_\mathrm{c} = 0$
$t = \epsilon /$
$t := 0, u''_\mathrm{c} \in \mathbf{u}$

Figure 4: Hybrid automaton defining system input $u''_\mathrm{c}$ from the bounded input set $\mathbf{u}$. Inputs are updated nondeterministically with period $\epsilon$ with respect to clock $t$.

directly in a verification tool, the hole can alternatively be modeled as shown in Figure 4.

## 2.4 Configurability

The model can be instantiated with varying number of discrete states and degrees of nondeterminism. The following table summarises the configurability of the model.

| | | | |
|---|---|---|---|
| *Initial values* | deterministic (all variables set to 0) | $\psi \in [-a, a]$ | nondeterministic for all variables |
| *Plant model* | linear | nonlinear | |
| *Controller model* | discrete vs. continuous time controller | linear vs. hybrid model (saturation) | |
| *Hole model* | as additive uncertainty | as multiplicative uncertainty | as hybrid automaton |

The simplest configuration is given by *deterministic initial values, deterministic plant, continuous time controller, hole as additive uncertainty.* The corresponding hybrid automaton consists of a single discrete state without any transition. The only source of nondeterminism in this model is introduced by the hole. Other configurations also involve nondeterministic transitions, e.g. *discrete time controller* and *hole as hybrid automaton.*

## 3 Verification challenge

The verification challenge is: given a model configuration, determine if $\psi$ is bounded in all reachable states under bounded system inputs $u''_{\mathrm{c}}$. Moreover, it would be interesting to construct the most permissive hole such that $\psi$ remains bounded. Note that the bound should be reasonable, i.e. $\psi \in [-\frac{\pi}{2} + \epsilon, \frac{\pi}{2} - \epsilon]$ for some suitable $\epsilon > 0$. Simulations of the nonlinear plant and linear controller model indicate that a reasonable bound is $\psi \in [-\frac{\pi}{2.26}, \frac{\pi}{2.26}]$ and for the nonlinear plant and saturated controller model it is $\psi \in [-\frac{\pi}{10.86}, \frac{\pi}{10.86}]$. The difficulty of the benchmark can be increased by replacing the linearised plant model by the actual nonlinear model or an approximation via multiple linearisation points.

## A Appendix

Using the default parameters described in the NXTway-GS documentation, the matrices occurring in the equations of the linearised plant and controller model are as follows.

$$A_{\mathrm{p}} = \begin{pmatrix} -162.1272879 & 0 & 162.1272879 & -409.7184124 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 78.14959139 & 0 & -78.14959139 & 269.6273393 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -92.4134822 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$B_{\mathrm{p}} = \begin{pmatrix} 157.5798419 & 157.5798419 \\ 0 & 0 \\ -75.95760351 & -75.95760351 \\ 0 & 0 \\ -51.3265211 & 51.3265211 \\ 0 & 0 \end{pmatrix}$$

$$C_{\mathrm{p}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 57.29577951 & 0 & -57.29577951 & 0 & -100.2676141 \\ 0 & 57.29577951 & 0 & -57.29577951 & 0 & 100.2676141 \end{pmatrix}$$

$$D_{\mathrm{p}} = \begin{pmatrix} 0.08087 & 0 \\ 0 & 0.08087 \end{pmatrix}$$

$$A_{\mathrm{c}} = \begin{pmatrix} 0 & 1.0 & 0 & -1.0 & 0 \\ 0 & 0 & 0.996 & 0 & 0 \\ 0 & 0 & -0.9999999525 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 49.99999763 & -49.99999763 \end{pmatrix}$$

$$B_{\mathrm{c}} = \begin{pmatrix} 0 & -0.00872664619 & -0.00872664619 & 0 & 0 \\ 0 & 0 & 0 & 0.0003 & 0 \\ 0 & 0 & 0 & 0.07499999644 & 0 \\ 0.999999992 & 0 & 0 & 0 & 0 \\ 0 & 0.4363322888 & 0.4363322888 & 0 & 0 \end{pmatrix}$$

$$\hat{A}_{\mathrm{c}} = \begin{pmatrix} 1 & 0.00400000019 & 0 & -0.00400000019 & 0 \\ 0 & 1 & 0.003984000189 & 0 & 0 \\ 0 & 0 & 0.996 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{pmatrix}$$

$$\hat{B}_{\mathrm{c}} = \begin{pmatrix} 0 & -0.00003490658642 & -0.00003490658642 & 0 & 0 \\ 0 & 0 & 0 & 0.000001200000057 & 0 \\ 0 & 0 & 0 & 0.0003 & 0 \\ 0.004000000158 & 0 & 0 & 0 & 0 \\ 0 & 0.001745329238 & 0.001745329238 & 0 & 0 \end{pmatrix}$$

$$C_{\mathrm{c}} = \begin{pmatrix} -5.529986398 & -10.31821442 & -13.529195 & 1161.547315 & -679.1764238 \\ -5.529986398 & -10.31821442 & -13.529195 & 1161.547315 & -679.1764238 \end{pmatrix}$$

$$D_{\mathrm{c}} = \begin{pmatrix} 43.95659666 & 6.016975757 & 6.016975757 & -0.004075058736 & 0.25 \\ 43.95659666 & 6.016975757 & 6.016975757 & -0.004075058736 & -0.25 \end{pmatrix}$$