Automatic Control Laboratory, ETH Zürich
Profs. M. Morari, J. Lygeros

Manual prepared by: Stefan Richter, Claudia Fischer, Aldo Zgraggen
Revision from: September 13, 2013

IfA Fachpraktikum - Experiment 1.3:

Chügelimat

In this experiment different balls are moved through a "ball exchange machine". The control logic which has to be developed is coded in Stateflow. Stateflow is a development tool based on Matlab/Simulink where state charts are drawn graphically and subsequently directly compiled. This allows a shorter control development time as compared with direct procedural logic languages (C/C++).

For the preparation of this experiment at home you need to download the following files from the IfA Fachpraktikum website `http://control.ee.ethz.ch/~ifa-fp/` :

`stateflow-tutorial.pdf`	introduction to the Stateflow software
`GettingStartedWithSimulink.pdf`	introduction to the Matlab tool Simulink
`chuegelimat.zip`	Simulink model of the Chügelimat

During the lab session you will need to extend a provided simulink model:
`chuegelimat.mdl` Simulink model of the Chügelimat

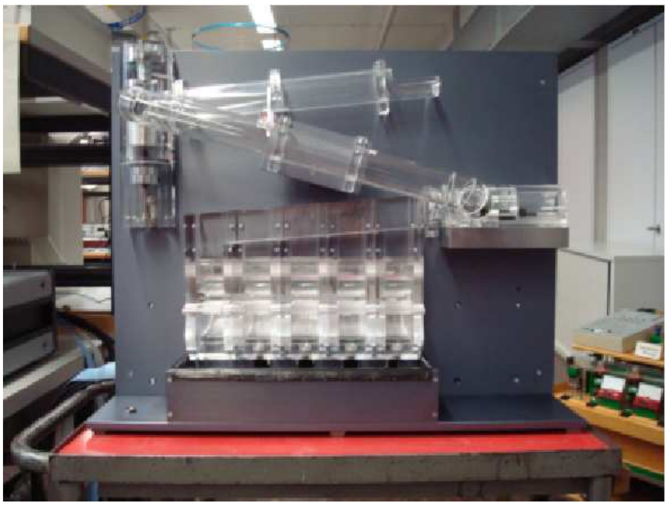

Figure 1: "Ball Exchange Machine" - Chügelimat

Contents

Chapter 1

Introduction

The automation of industrial processes can often be accomplished through structural logic design or "rule" control.
With Stateflow a tool is available in which it is possible to create state charts in an intuitive, graphical way and compile them directly. Stateflow is running in the Matlab/Simulink environment. In order to complete the lab practice efficiently, it's advisable to first read the

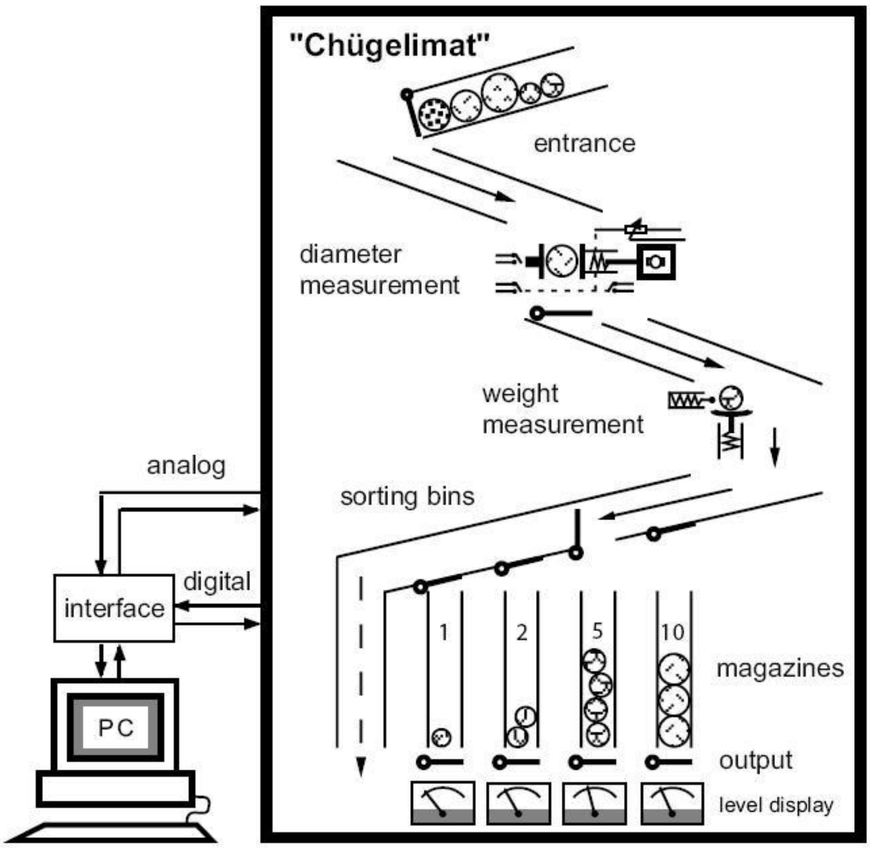

Figure 1.1: Schematic of the Chügelimat

"Introduction to Stateflow" and the "Getting Started with Simulink" manuals available on internet.
Stateflow is used in this experiment to control a model of a "coin exchange machine" named Chügelimat (Chügeli is Swiss German and means little ball), where balls are used instead of

coins. Balls with values of 1, 2, 5 and 10 are acceptable and appropriate exchange stock is handed out, however Chügelimat should ignore invalid balls.
The electromechanical setup of the Chügelimat is shown in figure 1.1. During the exchange procedure the ball must go through different stations of Chügelimat: the entrance, the diameter measurement station, the weight measurement station (balance) and the sorting bins station. In the sorting bins station a valid ball is stored in one of the four magazines (1, 2, 5 and 10) and if possible exchange stock is handed out (see table 1.1). If insufficient exchange stock is available

Value of thrown in ball	Value of ejected ball
1 x 1er	1 x 1er
1 x 2er	2 x 1er
1 x 5er	2 x 2er + 1 x 1er
1 x 10er	2 x 5er

Table 1.1: Ball classification at sorting bins

in the magazines or an invalid ball is thrown in, then the ball is ejected.
In the experiment, the processes for the entrance and the specification of the ball (diameter and weight measurement) should be coded in Stateflow. The remaining processes (sorting and exchanging the ball) to complete the control of Chügelimat already exist and have to be synchronized with the written processes. In the following sections the processes are explained more precisely. After this lab practice you will:

- have designed a control for a simple process, and
- know the possibilities and limits of modern, graphical software tools.

Chapter 2

Problem Setup and Notation

The electromechanical setup of the Chügelimat, shown in figure 1, consists of the following functional groups:

1. entrance
2. diameter measurement (DMS)
3. weight measurement (balance)
4. sorting bins
5. magazines

The processes for the first three functional groups, namely, the entrance, DMS and balance sections should be coded in Stateflow. These functional groups are described in detail in the following sections.
The notations used in this chapter are the same as those in the prepared Stateflow model. All digital signals to and from Chügelimat work with logic levels where 0 = off and 1 = active.
The goal of this project is to implement the three functional groups “entrance”, “DMS”, and “balance” and synchronize them to the existing units as described in the following sections.

2.1 Entrance

The slot where balls can be entered into Chügelimat is called “entrance”. A light barrier is used to detect when a ball is present and can be entered into the system. Table 2.1 illustrates the

Sensor or actuator	Stateflow variable	Comment
Light barrier (sensor)	inEntrance	Activated approx. 1s after ball is detected by light barrier. Deactivated as soon as ball is thrown out of entrance.
Entrance gate (actuator)	ejectEntrance	At activation the ball rolls to DMS section.

Table 2.1: Sensors and actuators at entrance

relationship between sensor and actuator signals of Chügelimat and the corresponding Stateflow variables.

The entrance gate should open when there is

1. a ball in front of the entrance, and
2. there is no ball in the diameter measurement station.

Hint: You should introduce a variable to determine if there is a ball in the DMS section, which may be set to 1 if there is a ball and reset to 0 if there is no ball.

2.2 Diameter Measurement Station (DMS)

This section describes the diameter measurement station. In figure 2.1 the setup of the DMS station is illustrated schematically. Three inductive sensors and two motors are used in this station. The sensors are on as soon as they are in the same height with a metal plate. The ball diameter is measured in terms of time.
Initially the lower plate of the bar driven by the stepper motor is at the level of the sensor at

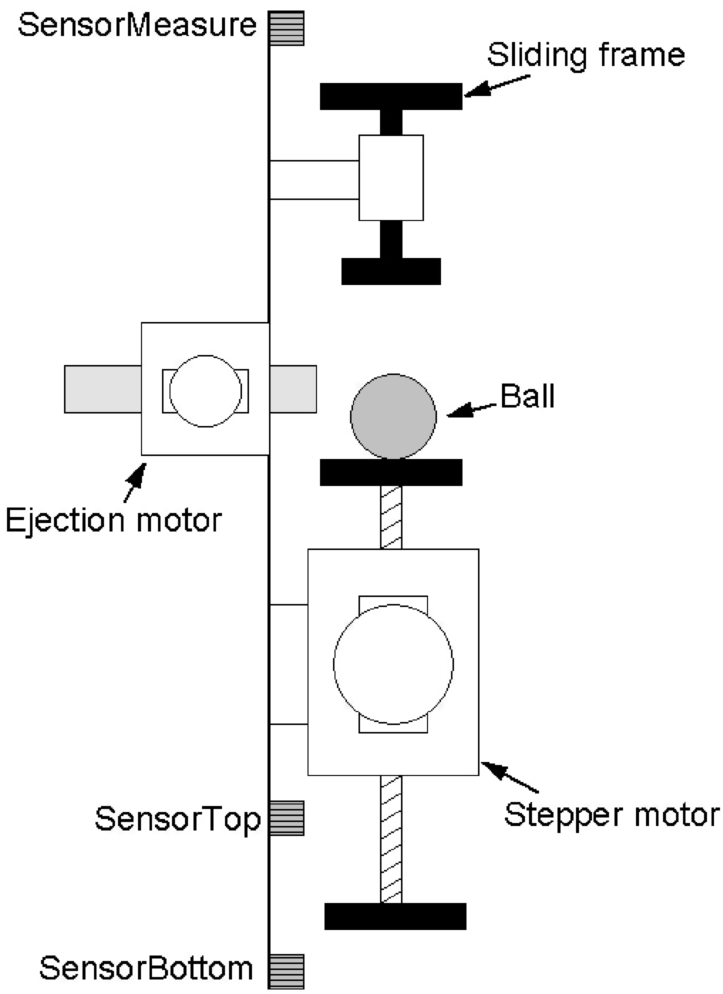

Figure 2.1: Side view of the diameter measurement station (DMS)

bottom and hence the *sensorBottom* is on. When the ball enters the DMS section the stepper motor moves the ball upwards. The sliding frame is pushed upwards until the *sensorMeasure* is turned on by the upper plate of the sliding frame. The time elapsed from the start of the stepper motor's operation to the instance when *sensorMeasure* is on should be assigned as the diameter of the ball. As soon as the *sensorMeasure* is on stepper motor moves downwards until the *sensorBottom* is on again. When the *sensorBottom* is on the ball is ejected by the ejection motor. *SensorTop* is used to stop the stepper motor in case it moves upwards without a ball.
In order to take correct measurements, the motor must always be switched off when the *sensorBottom* is on. Table 2.2 illustrates the relationship between sensor and actuators signals of Chügelimat and the corresponding Stateflow variables.

Ejection out of the diameter measurement station is allowed, if:

Sensors	Stateflow variable	Comments
SensorBottom	*sensorBottom*	Activated when a metal plate is detected.
SensorTop	*sensorTop*	Activated when a metal plate is detected.
SensorMeasure	*sensorMeasure*	Activated when a metal plate is detected.
Diameter	*diameter*	The time elapsed from the start of the motor's upwards motion until the sensorMeasure is on should be assigned as the diameter of the ball.
Actuators	**Stateflow variable**	**Comments**
Ejector DMS	*ejectDMS*	Activation causes the ball to be ejected.
Stepper motor on/off	*motorEnable*	0 ⇔ motor off, 1 ⇔ motor on
Stepper motor direction	*motorDirection*	0 ⇔ upwards, 1 ⇔ downwards

Table 2.2: Sensors and actuators of the DMS station

Sensors or actuators	Stateflow variable	Comments
Weight sensor	*weightMeasured*	This analogue input can be read in by Stateflow. as soon as a ball is on the balance.
Ejector	*ejectBalance*	Activation causes the ball to be ejected.

Table 2.3: Sensors ans actuators of balance unit

1. a ball is in the diameter measurement station,
2. the *sensorBottom* is on, and
3. there is no ball in the balance area.

Hint: You should introduce a variable to determine if there is a ball in the balance section, which may be set to 1 if there is a ball and reset to 0 if there is no ball.

2.3 Balance

Here, the weight of the ball is determined. The value can be read in by Stateflow via the "analogue" variable *weightMeasured*.
Table 2.3 illustrates the relationship between sensor and actuator signals of Chügelimat and the corresponding Stateflow variables.

Ejecting a ball is allowed, if

1. a ball is in the balance,
2. the sorting transaction of the last ball is finished, and
3. the sorting flaps in the next section are set.

To fulfill the last two conditions the program has to communicate with the sort algorithms already implemented in Stateflow. This communication is treated in the next section.

2.4 Interface to Chügelimat Framework

A part of the flow chart to control Chügelimat, containing the sorting and the balance sensor and stepper motor initialization algorithms is already prepared (see figure 2.2). For communication with the corresponding processes the following variables and events have been globally predefined in the Stateflow chart.

Figure 2.2: Prepared framework in Stateflow with initialization and simple sort algorithms

- **setFlaps (event):** Must be generated as soon as the ball weight is determined. Required by the sort algorithm as it initiates the setting of the sort flaps.

- **flapsPositioned (event):** Generated by the sort algorithm as soon as the sort flaps are positioned for a measured ball in the balance.

- **diameter (variable):** Diameter determined in the DMS station must be stored in this variable.

- **weight (variable):** Weight determined in the balance must be stored in this variable.

2.5 Magazines

The magazines have LED sensors installed to verify if a ball is available or not. This can be used to return correct exchange money/balls. This will be an additional task to implement during the lab session. For this purpose you will need the *magazinoutX* variables, where $X \in [1, 4]$, to throw out balls from the magazin. Additionally the Magazines LED information has to be fed into the stateflow chart.

Chapter 3

Preparation@Home

Task 1: Reading

Read the "Introduction to Stateflow". Users unpracticed in Simulink should read "Getting Started with Simulink".

Task 2: Understanding the Chügelimat framework

Make yourself familiar with the framework (see figure 2.2). Try to understand the flow of a ball passing through Chügelimat (i.e. which signals occur, what commands must be generated). The framework can be downloaded from the IfA-homepage.

Task 3: Control design

Sketch a possible state chart to control the entrance, the diameter and the weight measurement. Consider the specifications in the previous sections.
Additional hints:

- The three processes should run parallel.
- Ensure that the *sensorBottom* is ON in the DMS station when a ball is rolling in (especially for the first ball).
- Be careful to overwrite the values for diameter and weight of the old ball with the values of the new ball.
- Use the variables and events defined in section 2 to communicate with the framework.
- Stateflow syntax:
  - Place conditions and comparisons on transitions.
  - Place assignments inside the states of the flow chart.
  - The opposite is not implicitly prohibited but in practice causes errors.

Chapter 4

Lab Session Tasks

4.1 Preparations

To get started, double click the desktop link called "Ifa 1.3 Chuegelimat" on the lab PC. This will start Matlab, and download all the necessary framework files to C:/Scratch/Chuegelimat. Open the Simulink model "chuegelimat.mdl" which contains the control blocks. This is the file in which you will write your new control algorithms (beside the existing ones Sort/, Balanceinit/ and Motorinit).
Before you get started, make sure the DAQ card is turned on.

4.2 Experiment

Task 4: Implement Controller

Implement your processes for the entrance, the diameter and the weight measurement!
Attention: While implementing your algorithm, the stepper motor may get stuck. This may damage the motor severely. Whenever that happens you should stop the simulation and reconnect to the target (icon next to run/stop). This will stop the motor
Hints:

- "Introduction to Stateflow" includes information on how to define new variables.
- Build the model **after each coded process** to eliminate errors subsequently. Use the build button in the simulink file not the one in the stateflow chart. You should see the compilation report in the Matlab command window.

Once the Chügelimat is running, put different balls at the entrance and check if they are sorted properly. The valid balls are given in figure 4.1 and table 4.1. They are indicated with green stickers in the lab space.

Task 5: Clear Magazines (optional)

Once you have a working controller it is time to improve the working tasks. If you still have time try to integrate the LED sensor information to the stateflow chart

Value	Material	Diameter	Weight	Colour
1	Aluminium	$20mm$	$11.4g$	blue
2	Steel	$20mm$	$32.6g$	silver
5	Aluminium	$25mm$	$22.1g$	silver and yellow
10	Aluminium	$30mm$	$38.3g$	red

Table 4.1: Data of valid balls

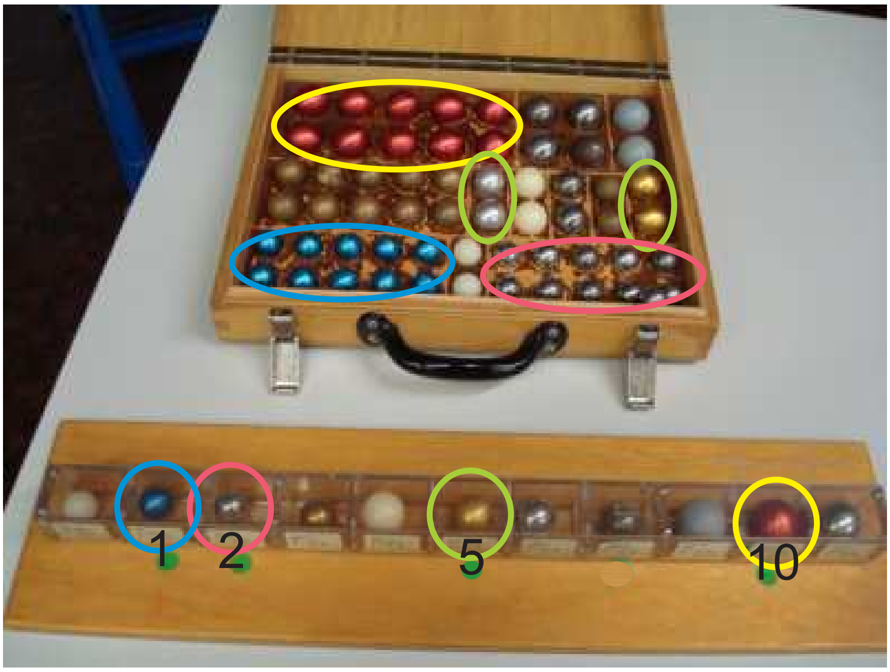

Figure 4.1: Valid balls for the Chügelimat.

and create a different version of the sorting state. The controller should check if it has enough coins/balls in its magazines to return the correct sum. If not then the given ball should be thrown out.

Task 6: Completion of Experiment

Please, fill out the online feedback form on the registration page under `MyExperiments`. Each student/participant has to fill out its own feedback form. This will help us to improve the experiment. Thank you for your help.