

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL ENGINEERING

DEPARTMENT OF CONTROL ENGINEERING



BACHELOR WORK

Logical processes modeling and optimization in Breweries

Staropramen

Prague, 2006

Peter Turza

Declaration

I declare, that I have worked out this bachelor work by myself and I have used only references (literature, projects, SW etc.) listed in the attached index.

In Prague, _____

signature

Acknowledgement

I want to thank my bachelor work leader, Ing. Jirka Roubal, for his excellent guidance and useful comments. I also want to thank to the employees of Staropramen breweries for their willingness to provide the data needed to finish this work.

Abstract

This work deals with modeling of logistic processes in the Staropramen brewery. Its outcome is supposed to be a model, which includes only the primary logistic processes. The model is expected to be prepared for a controller, which will enable the optimization of these processes.

Table of content

Chapter 1 Introduction	1
Chapter 2 Brewery structure analysis	3
2.1 Locations of facilities in the Czech Republic	3
2.2 Brief description of the facilities	4
2.2.1 Prague	4
2.2.2 Ostrava and Olomouc	4
2.3 Logistic between the facilities	4
2.4 Model objectives	5
Chapter 3 Theoretical introduction	6
3.1 State space equations of continuous linear systems	6
3.2 Basic relations between systems.....	7
3.2.1 Parallel connection of subsystems	8
3.2.2 Serial connection of subsystems	10
3.2.3 Feedback connection of subsystems	11
3.3 Matlab Simulink description	13
3.3.1 Step block	14
3.3.2 Gain block	14
3.3.3 Integrator block.....	14
3.3.4 Sum block	15
3.3.5 Product block.....	16
3.3.6 Transport delay block	16
3.3.7 Display block	17
3.3.8 From and goto blocks	17
Chapter 4 Construction of the model	19
4.1 Control inputs	19
4.1.1 Production line operation switch	19
4.1.2 Returnable container usage.....	20
4.1.3 Transport.....	21

4.2 Production facility Prague – Smíchov	25
4.2.1 Automated production line	25
4.2.2 Finished products stock	28
4.2.3 Raw – material stock	30
4.3 Central stock Prague – Radotín	31
4.3.1 Final products stock.....	32
4.3.2 Raw – material stock Prague – Radotín.....	33
4.3.3 Finalized model	34
Chapter 5 Model verification.....	37
5.1 Inputs and requested actions.....	37
5.2 Expected results of the model verification	38
5.3 Results of the model verification.....	39
Chapter 6 Introduction to optimization	43
6.1 Optimization requests	43
6.2 Model predictive control	43
Chapter 7 Final evaluation	45
References.....	47
Appendix A.....	48
Appendix B.....	49

Chapter 1

Introduction

The goal of this bachelor work is to create a model which will simulate logistic processes in the Staropramen brewery. Outputs of the model shall be the informations representing the capacity to which the stocks are filled. This is supposed to help the brewery's logistic-planning department manage and optimize product flows between their production facilities and stocks.

The main requirement is to make the model variable and expandable for future changes in the brewery. The model will be also the basic element on which the optimization of logistic processes will be based, thus the inputs and the structure of the model have to be prepared for a process controller.

To achieve this goal, it is necessary to collect information and data about the logistic processes between the individual objects of the brewery, consider their importance, understand how these processes affect each other and depend on each other and find the most accurate mathematical representation of these relations.

In order to make the model flexible, it will be split into several parts. Each part will be a stand-alone model, but these parts can be formed into the complex model very easily, by simply connecting their I/O ports. This way, when parameters and dependances between them change, it is not necessary to reconstruct the whole model. An adjustment in the relevant part of the model is sufficient.

Mathworks' Matlab Simulink was chosen as the environment in which the model should be built. This choice has been made because of Simulink's simplicity and especially clearness, which is important in a project of this extent. Another advantage of this choice

is, that the follow-up to this project, the optimization of the logistic processes, can be very easily merged into one model. After verifying the model, it can be easily reprogrammed by using any of the commonly used programming languages. This gives the brewery the most affordable and simplest way to use the model.

This bachelor work can be split into seven chapters. In chapter 2, following this introduction, a basic structure analysis of the brewery is done. Based on this analysis, the theoretical information needed to construct the model is provided in chapter 3. Chapter 4 focuses on the actual construction of the model, which is then verified in chapter 5. A short introduction to model predictive control, which will be used in the follow up work, is provided in chapter 6. This work is ended with a final conclusion, where there are discussed the problems relating to this work and their solutions.

Chapter 2

Brewery structure analysis

A structure analysis of the brewery must be done in order to construct the model. Since it is expected to be the model of the logistics, the most important thing is to analyse the connections between the facilities and the material flow between them.

2.1 Locations of facilities in the Czech Republic

The Staropramen brewery is a part of multinational Inbev corporation, which incorporates the world's largest breweries.

The Staropramen brewery has its main facilities located in Prague, Ostrava and Olomouc (fig. 2.1). Main facilities include production facilities and primary stocks.

Other facilities include secondary (external) stocks, and these are spread over the Czech Republic to cover the whole country. Overall, there are 15 secondary stocks.



Fig. 2.1: Locations of main facilities in the Czech Republic

Table 2.1: Distances between the main facilities

	Prague	Ostrava	Olomouc
Prague	-	380 km	280 km
Ostrava	380 km	-	100 km
Olomouc	280 km	100 km	-

2.2 Brief description of the facilities

A brief description of the facilities is needed to help understand the structure of logistic processes in the brewery.

2.2.1 Prague

In Prague, there are currently two production facilities. One is located in Prague – Smíchov, the other one is located in Praha – Braník. Each of these production facilities has two own stocks. One to store the final products until they are moved to central stock and one to store the raw material used during production.

The central stock of the brewery is located in Prague – Radotín.

2.2.2 Ostrava and Olomouc

There is one production facility in Ostrava with the same structure as the production facilities in Prague. There is a primary stock located in Olomouc.

2.3 Logistic between the facilities

The block diagram (fig 2.2) shows the material flow between the brewery's facilities. Raw material is transported from the stock labeled as CS – central stock, where it is stored, to the production facility and there it is stored in a stock labeled as RS – raw material stock. Finished products, which are temporarily stored in the stock labeled as PS – product stock, are transported to the central stock. From the central stock, products are transported to stocks labeled as SS – secondary stock. From here, they are transported to stores and their stocks labeled as C – customers. The stores then send back returnable containers, which are afterwards kept in the central or secondary stock, depending on current and forecasted needs of the brewery.

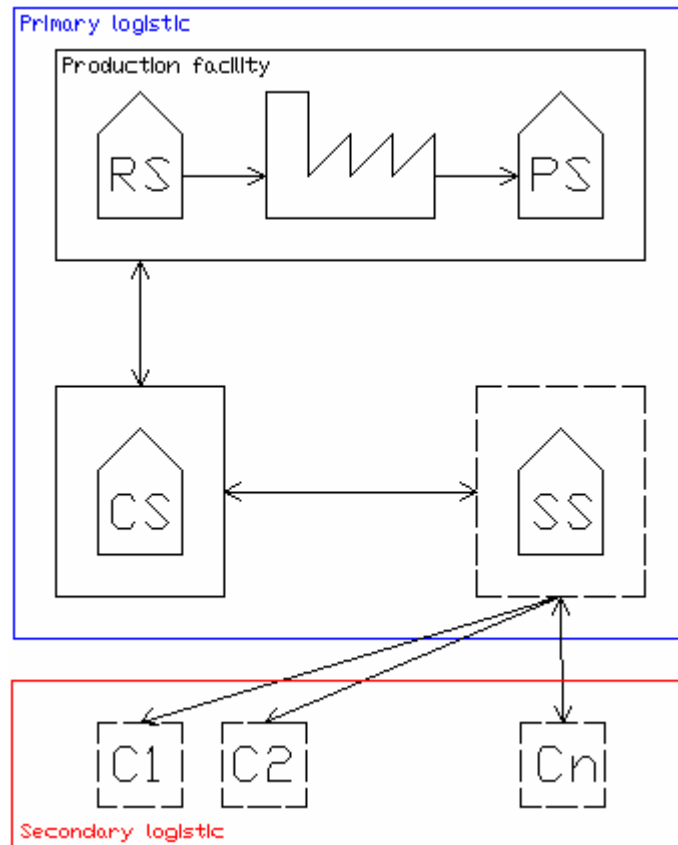


Fig 2.2: Material flow between facilities

2.4 Model objectives

The model resulting from this bachelor work is expected to cover only the primary logistic between the facilities located in Prague. This does not include the Braník production facility, because of it being closed down soon.

After receiving feedback from Staropramen logistic – planning department, the model shall be revised and adjusted to cover their needs more precisely. This, along with the optimization of the logistic processes, shall be the content of the follow-up diploma work.

Chapter 3

Theoretical introduction

Theoretical information provided in this chapter is expected to cover all areas needed to construct the model. It is needed to know the basics of system modeling and how the real-world system can be represented using a computer interface (Simulink).

3.1 State space equations of continuous linear systems

State space description (fig. 3.1) represents the dynamic behaviour of the relations between the input quantities (vector $\mathbf{u}(t)$ with dimension r), output quantities (vector $\mathbf{y}(t)$ with dimension m) and state quantities (vector $\mathbf{x}(t)$ with dimension n).

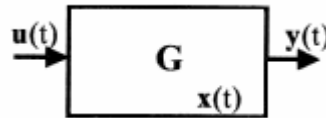


Fig. 3.1: State space description of dynamic system

State space description of a system can be generally expressed by a form of state and output equation

$$\begin{aligned} \mathbf{x}(t) &= \boldsymbol{\varphi}(t, \tau, \mathbf{x}(\tau), \mathbf{u}(\tau, t)) \\ \mathbf{y}(t) &= \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t)) \end{aligned} \quad (3.1)$$

where $\boldsymbol{\varphi}$ is a transfer function of a state, which expresses how the state $\mathbf{x}(\tau)$ changes in time τ , by the action of control (input) signal $\mathbf{u}(v)$ on the interval $v \in (\tau, t)$, to state $\mathbf{x}(t)$ in time t , \mathbf{g} is the transformation of momentary state $\mathbf{x}(t)$, control $\mathbf{u}(t)$ and time t to output $\mathbf{y}(t)$.

Because of limitation only to continuous differential dynamic systems, it is possible to express the trend of the state by a differential equation

$$\dot{\mathbf{x}}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t), \mathbf{u}(t)) \quad (3.2)$$

When considering linear systems, the transfer function $\boldsymbol{\varphi}$ is linear with regard to initial state and control and output function \mathbf{g} is linear with regard to momentary value of the state and control. Thus it is possible, in a chosen base of the linear space, to express them by using matrixes. Then the state space description can be expressed by using state equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(t) \mathbf{x}(t) + \mathbf{B}(t) \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(t) \mathbf{x}(t) + \mathbf{D}(t) \mathbf{u}(t) \end{aligned} \quad (3.3)$$

where $\mathbf{A}(t)$ is the matrix of the system with dimension $n \times n$, $\mathbf{B}(t)$ is the control matrix with dimension $n \times r$, $\mathbf{C}(t)$ is the output matrix with dimension $m \times n$, $\mathbf{D}(t)$ is the output matrix with dimension $m \times r$.

Generally, the $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\mathbf{C}(t)$, $\mathbf{D}(t)$ are continuous-time functions. If considering stationary systems, the state equations can be expressed as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \mathbf{x}(t) + \mathbf{D} \mathbf{u}(t) \end{aligned} \quad (3.4)$$

where the matrixes \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} are constant [1].

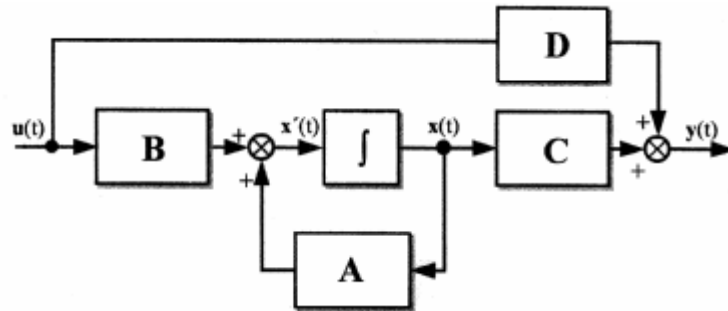


Fig. 3.2: Block diagram of continuous linear dynamic system

3.2 Basic relations between systems

Let a dynamic system be a combination of two subsystems. The dynamic behaviours of these subsystems can be expressed by using transfer function or state space description.

Two multidimensional systems S_1 and S_2 can be connected by using:

- parallel connection
- serial connection
- feedback connection

Let S_1 be a system described by the following state equations

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u}_1 \\ \mathbf{y}_1 &= \mathbf{C}_1 \mathbf{x}_1 + \mathbf{D}_1 \mathbf{u}_1\end{aligned}\quad (3.5)$$

where $\mathbf{u}_1(t)$ is the vector of input quantities with dimension r_1 , $\mathbf{y}_1(t)$ is the vector of output quantities with dimension m_1 and $\mathbf{x}_1(t)$ is the vector of state quantities with dimension n_1 .

The system S_1 is also described by transfer matrix $\mathbf{G}_1(s)$.

Let S_2 be a system described by the following state equations

$$\begin{aligned}\dot{\mathbf{x}}_2 &= \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u}_2 \\ \mathbf{y}_2 &= \mathbf{C}_2 \mathbf{x}_2 + \mathbf{D}_2 \mathbf{u}_2\end{aligned}\quad (3.6)$$

where $\mathbf{u}_2(t)$ is the vector of input quantities with dimension r_2 , $\mathbf{y}_2(t)$ is the vector of output quantities with dimension m_2 and $\mathbf{x}_2(t)$ is the vector of state quantities with dimension n_2 .

The system S_2 is also described by transfer matrix $\mathbf{G}_2(s)$.

Deduction of the transfer function description $\mathbf{G}(s)$ and of the state space description

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} \\ \mathbf{y} &= \mathbf{C} \mathbf{x} + \mathbf{D} \mathbf{u}\end{aligned}\quad (3.7)$$

of the system S can be done as it is shown in following sections.

3.2.1 Parallel connection of subsystems

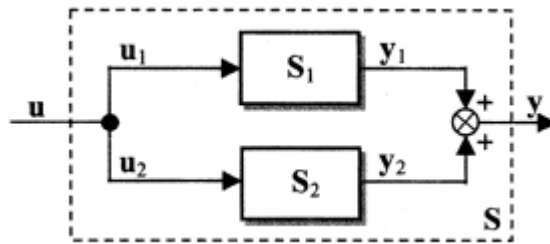


Fig. 3.3: Parallel connection of subsystems

If the systems S_1 and S_2 have to be connected with parallel connection, then the number of their inputs must be equal ($\dim \mathbf{u}_1 = \dim \mathbf{u}_2$) and also the number of their outputs must be equal ($\dim \mathbf{y}_1 = \dim \mathbf{y}_2$). Following statements apply to the resulting system S

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 = \mathbf{u}_2 \\ \mathbf{y} &= \mathbf{y}_1 + \mathbf{y}_2\end{aligned}\tag{3.8}$$

The transfer matrix $\mathbf{G}(s)$ of the system S can be expressed as

$$\begin{aligned}\mathbf{Y}(s) &= \mathbf{G}(s)\mathbf{U}(s) \\ \mathbf{Y}(s) &= \mathbf{Y}_1(s) + \mathbf{Y}_2(s) = \mathbf{G}_1(s)\mathbf{U}_1(s) + \mathbf{G}_2(s)\mathbf{U}_2(s) = [\mathbf{G}_1(s) + \mathbf{G}_2(s)]\mathbf{U}(s)\end{aligned}$$

or

$$\mathbf{G}(s) = \mathbf{G}_1(s) + \mathbf{G}_2(s)\tag{3.9}$$

Analogically, the transfer matrix $\mathbf{G}(s)$ of system S , which is created using parallel connections between subsystems $S_1 \dots S_n$, can be expressed as the sum of the transfer matrixes $\mathbf{G}_1(s) \dots \mathbf{G}_n(s)$ of the separate subsystems.

When deducting the state equations of the system S , the inputs \mathbf{u}_1 and \mathbf{u}_2 of the separate subsystems are replaced with input \mathbf{u} .

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} \\ \dot{\mathbf{x}}_2 &= \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u}\end{aligned}\tag{3.10}$$

Then the equation 3.9 can be written, using matrix notation, as

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}}_{\mathbf{B}} \mathbf{u}\tag{3.11}$$

The equation 3.10 is the state equation of the system S . The output equation of the system S can be deducted analogically

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1\mathbf{u}_1 = \mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1\mathbf{u} \\ \mathbf{y}_2 &= \mathbf{C}_2\mathbf{x}_2 + \mathbf{D}_2\mathbf{u}_2 = \mathbf{C}_2\mathbf{x}_2 + \mathbf{D}_2\mathbf{u}\end{aligned}\tag{3.12}$$

Output of the system S is equal to

$$\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2\tag{3.13}$$

Then the output equation of the system S can be written, using matrix notation, as

$$[\mathbf{y}] = \underbrace{[\mathbf{C}_1 \quad \mathbf{C}_2]}_{\mathbf{C}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \underbrace{(\mathbf{D}_1 + \mathbf{D}_2)}_{\mathbf{D}} \mathbf{u}\tag{3.14}$$

The state equation (3.11) and output equation (3.14) make the state description of the system S [1].

3.2.2 Serial connection of subsystems

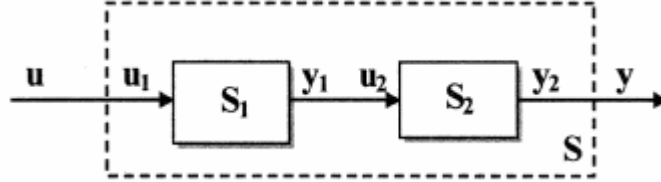


Fig. 3.4: Serial connection of subsystems

In order to make a serial connection between subsystems S_1 and S_2 , the S_1 subsystem's number of outputs must be equal to the S_2 subsystem's inputs ($\dim y_1 = \dim u_2$). Following statement applies to the resulting system S

$$\begin{aligned} y_1 &= u_2 \\ u &= u_1 \\ y &= y_2 \end{aligned} \quad (3.15)$$

In accordance with statements 3.14, the transfer matrix $G(s)$ of the system S can be expressed as

$$\begin{aligned} Y(s) &= G(s)U(s) \\ Y(s) = Y_2(s) &= G_2(s)U_2(s) = G_2(s)Y_1(s) = G_2(s)G_1(s)U_1(s) = \\ &= G_2(s)G_1(s)U(s) \end{aligned} \quad (3.16)$$

or

$$G(s) = G_2(s)G_1(s) \quad (3.17)$$

Analogically, the transfer matrix $G(s)$ of system S , which is created using serial connections between subsystems $S_1 \dots S_n$, can be expressed as the product of the transfer matrixes $G_1(s) \dots G_n(s)$ of the separate subsystems.

When deducting the state equations of the system S , the input u_1 in the state equation of subsystem S_1 is substituted with input u , the input u_2 in the state equation of subsystem S_2 is substituted with the output y_1 of the subsystem S_1 and input u_1 is set equal to input u

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 u_1 = A_1 x_1 + B_1 u \\ \dot{x}_2 &= A_2 x_2 + B_2 u_2 = A_2 x_2 + B_2 (C_1 x_1 + D_1 u) = A_2 x_2 + B_2 (C_1 x_1 + D_1 u) \end{aligned} \quad (3.18)$$

Then the equation 3.17 can be written, using matrix notation, as

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \underbrace{\begin{bmatrix} B_1 \\ B_2 D_1 \end{bmatrix}}_B u \quad (3.19)$$

The equation 3.18 is the state equation of the system S . The output equation of the system S can be deducted analogaly

$$\begin{aligned} y_1 &= C_1 x_1 + D_1 u_1 = C_1 x_1 + D_1 u \\ y_2 &= C_2 x_2 + D_2 u_2 = C_2 x_2 + D_2 y_1 = C_2 x_2 + D_2 (C_1 x_1 + D_1 u) = \\ &= D_2 C_1 x_1 + C_2 x_2 + D_2 D_1 u \end{aligned} \quad (3.20)$$

Output of the system S is equal to

$$y = y_2 = D_2 C_1 x_1 + C_2 x_2 + D_2 D_1 u \quad (3.21)$$

Then the output equation of the system S can be written, using matrix notation, as

$$[y] = \underbrace{\begin{bmatrix} D_2 C_1 & C_2 \end{bmatrix}}_C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{(D_2 D_1)}_D u \quad (3.22)$$

The state equation (3.18) and output equation (3.21) make the state description of the system S [1].

3.2.3 Feedback connection of subsystems

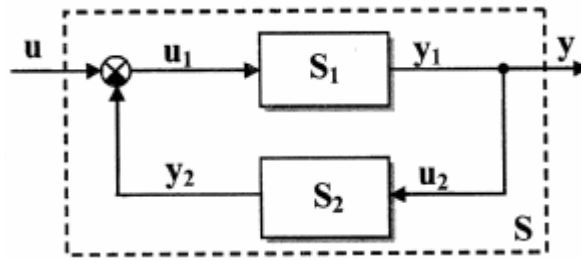


Fig. 3.5: Feedback connection of subsystems

In order to make a feedback connection between subsystems S_1 and S_2 , the S_1 subsystem's number of outputs must be equal to the S_2 subsystem's inputs ($\dim y_1 = \dim u_2$) and the S_2 subsystem's number of outputs must be equal to the S_1 subsystem's number of inputs ($\dim y_2 = \dim u_1$).

Following statement applies to the resulting system S

$$\begin{aligned} u_1 &= u \pm y_2 \\ y &= y_1 = u_2 \end{aligned} \quad (3.23)$$

The deduction of the transfer matrix $G(s)$ of the system is following

$$\begin{aligned} Y(s) &= Y_1(s) = G_1(s)U_1(s) = G_1(s)[U(s) - Y(s)] = \\ &= G_1(s)[U(s) - G_2(s)U_2(s)] = G_1(s)[U(s) - G_2(s)Y(s)] \end{aligned} \quad (3.24)$$

After applying algebraic modifications to equation 3.23, the equation changes to

$$Y(s) = [I + G_1(s)G_2(s)]^{-1} G_1(s)U(s) \quad (3.25)$$

The final transfer matrix $G(s)$ for the system S with negative feedback is

$$G(s) = [I + G_1(s)G_2(s)]^{-1} G_1(s) \quad (3.26)$$

For the system S with positive feedback, the final transfer matrix $G(s)$ is

$$G(s) = [I - G_1(s)G_2(s)]^{-1} G_1(s) \quad (3.27)$$

When deducting the state equations of the system S with a negative feedback, the first step is the deduction of the output equation using the equation 3.22 and state descriptions expressed by equations 3.5, 3.6 and 3.7.

$$\begin{aligned} y &= y_1 = C_1 x_1 + D_1 u_1 = C_1 x_1 + D_1(u - y_2) = C_1 x_1 + D_1(u - C_2 x_2 - D_2 u_2) = \\ &= C_1 x_1 + D_1 u - D_1 C_2 x_2 - D_1 D_2 y \end{aligned} \quad (3.28)$$

After applying algebraic modifications to equation 3.27, the equation changes to

$$y = (I + D_1 D_2)^{-1} (C_1 x_1 - D_1 C_2 x_2 + D_1 u) \quad (3.29)$$

The output equation of the system S with negative feedback can be written, using matrix notation, as

$$[y] = \underbrace{\left[(I + D_1 D_2)^{-1} C_1 \quad -(I + D_1 D_2)^{-1} D_1 C_2 \right]}_C \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \underbrace{(I + D_1 D_2)^{-1} D_1}_D u \quad (3.30)$$

The state equation deduction is done likewise

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 u_1 = A_1 x_1 + B_1(u - y_2) = A_1 x_1 + B_1 u - B_1(C_2 x_2 + D_2 u_2) = \\ &= A_1 x_1 + B_1 u - B_1(C_2 x_2 + D_2 y) = A_1 x_1 + B_1 u - B_1 C_2 x_2 - B_1 D_2 y \\ \dot{x}_2 &= A_2 x_2 + B_2 u_2 = A_2 x_2 + B_2 y \end{aligned} \quad (3.31)$$

By substituting y in output equation 3.29 with state equations 3.30, the state equation of the system S , using matrix notation, is following

$$\begin{aligned}
\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_x &= \underbrace{\begin{bmatrix} A_1 - B_1 D_2 (I + D_1 D_2)^{-1} C_1 & -B_1 C_2 + B_1 D_2 (I + D_1 D_2)^{-1} D_1 C_2 \\ B_2 (I + D_1 D_2)^{-1} C_1 & A_2 - B_2 (I + D_1 D_2)^{-1} D_1 C_2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \\
&+ \underbrace{\begin{bmatrix} B_1 - B_1 D_2 (I + D_1 D_2)^{-1} D_1 \\ B_2 (I + D_1 D_2)^{-1} D_1 \end{bmatrix}}_B u
\end{aligned} \tag{3.32}$$

For a system with positive feedback, we analogically get the following equation

$$\begin{aligned}
\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_x &= \underbrace{\begin{bmatrix} A_1 + B_1 D_2 (I - D_1 D_2)^{-1} C_1 & B_1 C_2 + B_1 D_2 (I - D_1 D_2)^{-1} D_1 C_2 \\ B_2 (I - D_1 D_2)^{-1} C_1 & A_2 + B_2 (I - D_1 D_2)^{-1} D_1 C_2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \\
&+ \underbrace{\begin{bmatrix} B_1 + B_1 D_2 (I - D_1 D_2)^{-1} D_1 \\ B_2 (I - D_1 D_2)^{-1} D_1 \end{bmatrix}}_B u
\end{aligned} \tag{3.33}$$

The output equation of the system S with positive feedback can be written, using matrix notation, as [1]

$$[y] = \underbrace{\begin{bmatrix} (I - D_1 D_2)^{-1} C_1 & (I - D_1 D_2)^{-1} D_1 C_2 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \underbrace{(I - D_1 D_2)^{-1} D_1}_D u \tag{3.34}$$

3.3 Matlab Simulink description

Simulink is a platform for multidomain simulation and model-based design of dynamic systems. It provides an interactive graphical environment and a customizable set of block libraries that allows accurately design, simulate, implement, and test control, signal processing, communications, and other time-varying systems.

Add-on products extend the Simulink environment with tools for specific modeling and design tasks and for code generation, algorithm implementation, test, and verification.

Simulink is integrated with Matlab, providing immediate access to an extensive range of tools for algorithm development, data visualization, data analysis and access, and numerical computation.

3.3.1 Step block



Fig 3.6: Symbol of step block

The Step block provides a step between two definable levels at a specified time. If the simulation time is less than the Step time parameter value, the block's output is the Initial value parameter value. For simulation time greater than or equal to the Step time, the output is the Final value parameter value.

The block's numeric parameters must be of the same dimensions after scalar expansion. If the Interpret vector parameters as 1-D option is off, the block outputs a signal of the same dimensions and dimensionality as the parameters. If the Interpret vector parameters as 1-D option is on and the numeric parameters are row or column vectors (i.e., single row or column 2-D arrays), the block outputs a vector (1-D array) signal; otherwise, the block outputs a signal of the same dimensionality and dimension as the parameters [5].

3.3.2 Gain block



Fig. 3.7: Symbol of gain block

The Gain block multiplies the input by a constant value (gain). The input and the gain can each be a scalar, vector, or matrix.

The value of the gain is specified in the Gain parameter. The Multiplication parameter lets specify element-wise or matrix multiplication. For matrix multiplication, this parameter also lets indicate the order of the multiplicands [5].

3.3.3 Integrator block



Fig. 3.8: Symbol of integrator block

The Integrator block outputs the integral of its input at the current time step. The following equation represents the output of the block \mathbf{y} as a function of its input \mathbf{u} and an initial condition \mathbf{y}_0 , where \mathbf{y} and \mathbf{u} are vector functions of the current simulation time t

$$\mathbf{y}(t) = \int_{t_0}^t \mathbf{u}(\tau) d\tau + \mathbf{y}_0 \quad (3.35)$$

Simulink can use a number of different numerical integration methods to compute the Integrator block's output, each with advantages in particular applications. The Solver pane of the Configuration parameters dialog box allows to select the technique best suited to the application.

Simulink treats the Integrator block as a dynamic system with one state, its output. The Integrator block's input is the state's time derivative

$$\begin{aligned} \mathbf{x} &= \mathbf{y}(t) \\ \mathbf{x}_0 &= \mathbf{y}_0 \\ \dot{\mathbf{x}} &= \mathbf{u}(t) \end{aligned} \quad (3.36)$$

The currently selected solver computes the output of the Integrator block at the current time step, using the current input value and the value of the state at the previous time step. To support this computational model, the Integrator block saves its output at the current time step for use by the solver to compute its output at the next time step. The block also provides the solver with an initial condition for use in computing the block's initial state at the beginning of a simulation run. The default value of the initial condition is 0. The block's parameter dialog box allows to specify another value for the initial condition or create an initial value input port on the block.

To prevent the output from exceeding specifiable levels, select the Limit output check box and enter the limits in the appropriate parameter fields. Doing so causes the block to function as a limited integrator. When the output reaches the limits, the integral action is turned off to prevent integral wind up [5].

3.3.4 Sum block



Fig. 3.9: Symbol of sum block

The Sum block performs addition or subtraction on its inputs. This block can add or subtract scalar, vector, or matrix inputs. It can also collapse the elements of a single input vector.

The operations of the block are specified with the list of signs parameter. Plus (+), minus (-), and spacer () characters indicate the operations to be performed on the inputs. A spacer character creates extra space between ports on the block's icon.

The Sum block first converts the input data type(s) to the output data type using the specified rounding and overflow modes, and then performs the specified operations [5].

3.3.5 Product block



Fig. 3.10: Symbol of product block

The Product block performs multiplication or division of its inputs. This block produces outputs using either element-wise or matrix multiplication, depending on the value of the Multiplication parameter. The operations are specified with the Number of inputs parameter. Multiply(*) and divide(/) characters indicate the operations to be performed on the inputs [5].

3.3.6 Transport delay block



Fig. 3.11: Symbol of transport delay block

The Transport Delay block delays the input by a specified amount of time. It can be used to simulate a time delay.

At the start of the simulation, the block outputs the Initial output parameter until the simulation time exceeds the Time delay parameter, when the block begins generating the delayed input. The Time delay parameter must be nonnegative.

When output is required at a time that does not correspond to the times of the stored input values, the block interpolates linearly between points. When the delay is smaller than the step size, the block extrapolates from the last output point, which can produce inaccurate results. Because the block does not have direct feedthrough, it cannot use the current input to calculate its output value [5].

3.3.7 Display block

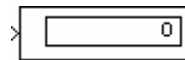


Fig. 3.12: Symbol of display block

The Display block shows the value of its input on its icon [5].

3.3.8 From and goto blocks



Fig. 3.13: Symbol of from block

The From block accepts a signal from a corresponding Goto block, then passes it as output. The data type of the output is the same as that of the input from the Goto block. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them. To associate a Goto block with a From block, enter the Goto block's tag in the Goto Tag parameter.

A From block can receive its signal from only one Goto block, although a Goto block can pass its signal to more than one From block.



Fig. 3.14: Symbol of goto block

The Goto block passes its input to its corresponding From blocks. The input can be a real- or complex-valued signal or vector of any data type. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them.

A Goto block can pass its input signal to more than one From block, although a From block can receive a signal from only one Goto block. The input to that Goto block is passed to the From blocks associated with it as though the blocks were physically connected. Goto blocks and From blocks are matched by the use of Goto tags, defined in the Tag parameter [5].

Chapter 4

Construction of the model

The task of model construction can be, in this case, split into two areas. Constructing control input models, which enable to set the desired values and pre-calculate certain values used in actual model of the brewery's logistic processes. The other area deals with construction of the logistic processes.

4.1 Control inputs

In this model, there are expected to be three types of control inputs:

- Production line operation switch
- Returnable container usage
- Truck loading and unloading

4.1.1 Production line operation switch

The automated production lines do not operate full time, they need to be stopped due to production plans and maintenance. The best way to simulate it in this project, is to have a signal source, whose output can be set to 1 (line on) or 0 (line off) at a chosen time. This signal source can be simulated in Simulink using two step blocks (section 3.3.1) and a sum block (section 3.3.4). The idea of this solution is to have two step-changing signals, from which one changes its value to 1 in time labeled as starting time – t_s and the other one changes its value to 1 in time labeled as stopping time – t_e . This second signal is subtracted

from the first one and this results in a rectangular signal which has value 1 (line on) during the requested time period.

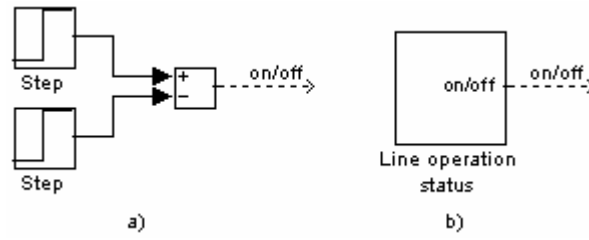


Fig. 4.1: On/off signal simulation – a) detailed model, b) masked model

When the line operation status block parameters are set to values $t_s = 2$ and $t_e = 8$, which means 2, respectively 8 hours from the start of the simulation, the signal behaviour is following:

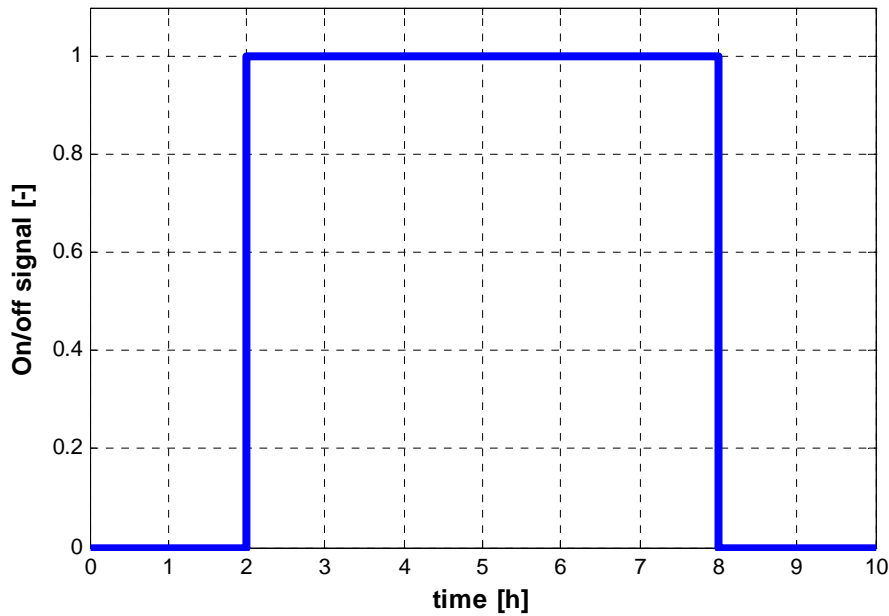


Fig. 4.2: On/off signal simulation

4.1.2 Returnable container usage

To ensure that returnable containers will be used only if it is requested by the production planning department, a returnable container usage control block must be included in the model.

Returnable containers are transported to the production lines only when production planning department is requesting it and if the specified automated line is in operation.

This leads to using a state control block described in section 4.1.1 with collaboration of a product block (section 3.3.5) as it is shown on figure 4.4.

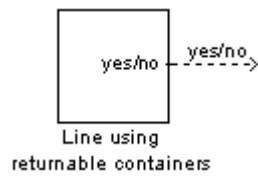


Fig. 4.3: Returnable container request usage block

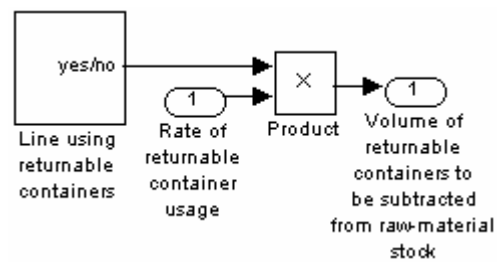


Fig. 4.4: Returnable container usage control block

4.1.3 Transport

Finished products and returnable containers (raw – material) are moved between facilities with trucks.



Fig. 4.5: Loading of a truck

The logistic planning department specifies, how many bottles and barrels (or returnable containers) are loaded and transported to another facility, so the model must allow to set these values and also the starting time of truck loading operation. Statistically was determined, that this operation takes 30 minutes to be completed. This time is labeled as truck loading time – T_T .

Then it is possible to determine the loading rate, at which the products (or returnable containers) are moved from the stock to the truck. Mathematically, this rate labeled as R_T , can be expressed as

$$R_T = \frac{N}{T_T} \quad (4.1)$$

Where N is the volume of the loaded cargo expressed in hectolitres.

In Simulink, this process can be represented as a truck loading (or unloading) control block based on the exact same principle as the two control blocks described in previous two sections (4.1.1 and 4.1.2), which is connected with a gain using serial connection. This gain expresses the loading rate (4.1). As a result, this subsystem has its output set to the loading (or unloading) rate until the truck is loaded (or unloaded). Then it is set to 0.

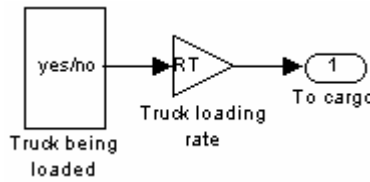


Fig. 4.6: Model of truck loading

The cargo space of the truck is represented by an integrator block (section 3.3.3). Its output value is changing depending on the rate, which is put through a sum block (section 3.3.4). If the rate is positive, then it represents loading and if it is negative, then it represents unloading. Mathematically the loaded volume, labeled as T_L can be expressed as

$$T_L = \int_{t_s}^{t_e} R_T d\tau \quad (4.2)$$

where t_s is the starting time of truck loading and t_e is the stopping time of truck loading.

After a truck is loaded, it carries the cargo to the specified location. This is represented by a transport delay. After it arrives at the specified location, the unloading process starts. The cargo is unloaded at the same rate as it was loaded (4.2) and relevant stock is filled at corresponding rate at the same time. Based on this fact the unloading

processes' rate and starting time can be determined from the loading time and loading rate. Only the transport delay value needs to be added to the starting time. Mathematically the unloaded volume, labeled as T_U can be expressed as

$$T_U = \int_{t_s+T_d}^{t_e+T_d} R_T d\tau \quad (4.3)$$

Where T_d is the transport delay between the specified locations.

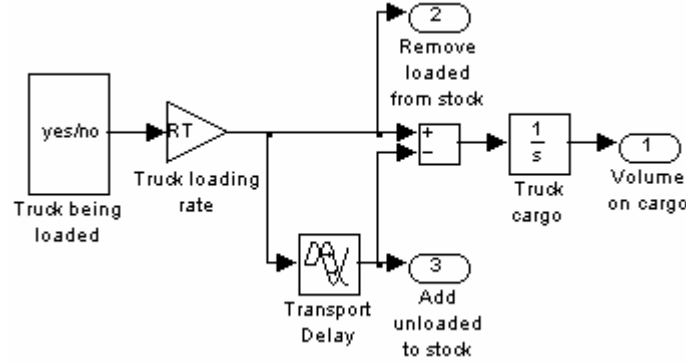


Fig. 4.7: General model of a truck



Fig 4.8: Masked general model of a truck

The output RTL, which represents the loading rate of the cargo, must be connected with the relevant stock input, from which the cargo is taken away. This way, it is subtracted from the stock. Analogically, the RTU output, which represents the unloading rate of the cargo, must be connected with the relevant stock input, to which the cargo is taken. This way, it is added to the stock. The principle of adding and subtracting products to and from stocks will be explained in the following sections.

If it is needed to keep track of different products loaded to the truck, the cargo area must be split into sections. Number of these sections is equal to the number of products on the truck. These sections are created by adding according number of general truck models

to the system. The loading time of the products can be set to equal value, since the truck leaves the facility only after the last product is loaded. This modification is shown on the following figures.

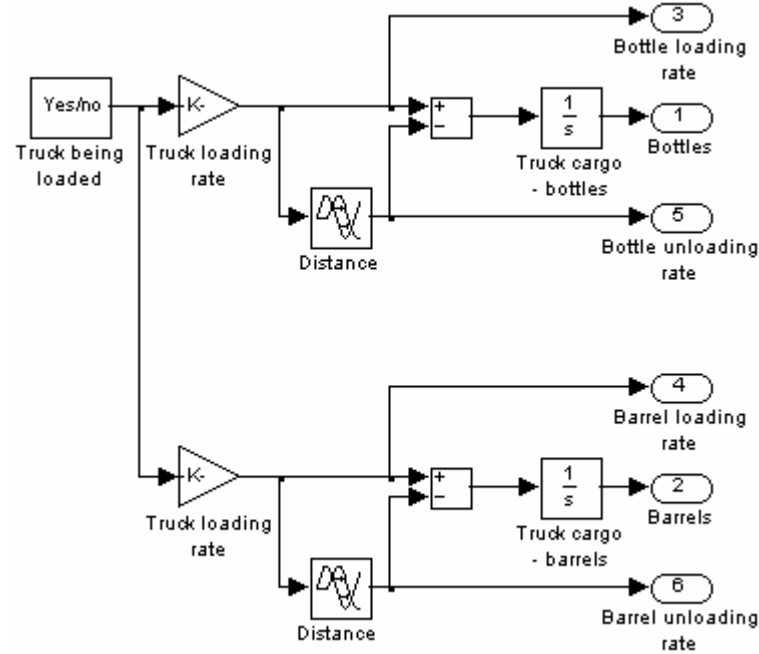


Fig. 4.9: General model of a truck with two cargo sections

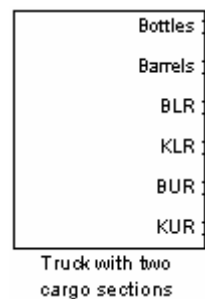


Fig. 4.10: Masked general model of a truck with two cargo sections

There are 6 logistic processes that have to be considered in this model. These processes are transport of finished products from production facility Prague – Smíchov and from secondary facilities to central stock Radotín, transport of returnable containers from central stock Radotín to production facility Prague – Smíchov and to secondary facilities, transport of returnable containers from secondary facilities to central stock Radotín and transport of finished products from central stock Radotín to secondary facilities. Therefore there are 6 truck subsystems needed to complete the model.

4.2 Production facility Prague – Smíchov

As it was suggested in section 2.2.1, the production facility in Prague – Smíchov includes the following objects:

- Three automated production lines
- Final product stock
- Raw material stock

4.2.1 Automated production line

An automated production line used in the Staropramen brewery is a mechanism, which is programmed to constantly perform the following actions:

- Taking clean empty containers (bottles or barrels) from a feeding machine
- Filling the containers with beer
- Closing the containers
- Labeling the containers
- Packing the containers

The empty containers can be either new bottles provided by a glass factory, or returnable containers delivered to Prague – Smíchov from central stock located in Prague – Radotín.

After the containers are packed, they are temporarily moved to production stock, where they are stored until they are transported to central stock in Prague – Radotín.

Since these actions are performed at a constant rate, given by the parameters of the automated line (table 4.1), and the goal of this work is to simulate logistic processes, it is possible to look at the automated line as on a transfer delay (section 3.3.6). This transfer delay represents the time, which is needed for a container to go through the whole process which was described above.



Fig. 4.11: Picture of the automated line located in Prague – Smíchov

Parameters of an automated production line are:

- State – turned on/off
- Maximum capacity – maximum amount of produced containers per hour
- Efficiency – capacity at which the automated line is actually running, given as a percentage from the maximum capacity
- Waste percentage – percentage from initial number of containers which can't be considered as finished products despite passing through the automated line.
- Transfer delay – time needed for a container to go through the automated production line

The values of these parameters are shown in table 4.1.

Table 4.1: Parameters of the automated production lines

Line number (type)	Max. capacity C [containers/hour]	Efficiency E [%]	Waste W [%]	Transfer delay T_D [hours]
1 (bottles)	24 000	60	0,1	1,5
2 (bottles)	24 000	70	0,1	0,5
3 (barrels)	300	90	0,1	0,5

These parameters express how many containers the automated line produces per time unit. This rate, labeled as R_L , can be expressed mathematically as

$$R_L = CE \left(1 - \frac{W}{100} \right) \quad (4.4)$$

In Simulink, this equation is represented as a serial connection of three gains, whose values are set to C , E , and $1 - \frac{W}{100}$.

After connecting these gains with the transfer delay, using serial connection, we get the model of the automated production line.

Because the Staropramen brewery has requested, that the values representing the stock status shall be evaluated in hectolitres instead of the number of containers or pallets, a conversion to this unit needs to be added. Since this value is the product of the number of containers and their volume, a gain block can be used for conversion. Its value is set to the volume value of the container. This block is then connected to the model of automated production line using serial connection.

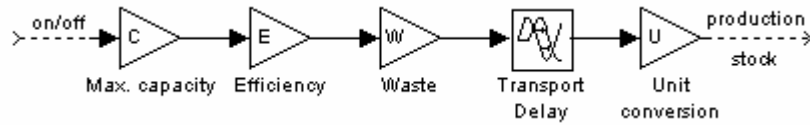


Fig 4.12: Model of an automated production line

The whole system of automated production lines, after connecting it to the source of controlling on/off signal (section 4.1.1) using serial connection is shown on following figures.

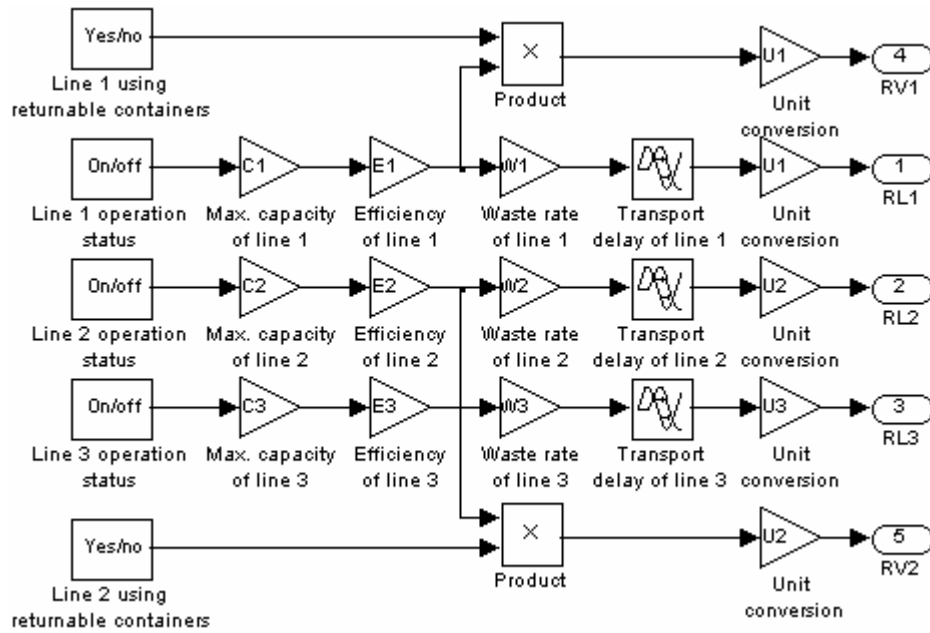


Fig. 4.13: Model of automated production lines located in Prague – Smíchov

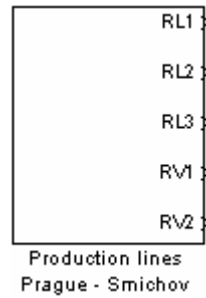


Fig. 4.14: Masked model of automated production lines located in Prague – Smíchov

4.2.2 Finished products stock

The Prague - Smíchov finished product stock is divided into two sections. One stores bottle products that come off the production lines 1 and 2 and the other one stores barrel products that come off the production line 3. The products are stored there temporarily, until they are moved with truck to other locations.



Fig. 4.15: Finished products stock in Prague – Smíchov

Since the rates at which the products come off the line and at which they are loaded to truck are constant, the volume of products stored in the first section of the stock, labeled as S_{SB} can be expressed mathematically as

$$S_{SB} = S_{SB0} + \int_{t_{s1}}^{t_{e1}} R_{L1} d\tau + \int_{t_{s2}}^{t_{e2}} R_{L2} d\tau - \int_{t_{ts}}^{t_{te}} R_T d\tau \quad (4.5)$$

where S_{SB0} is the initial volume of bottle products on the stock (expressed in hectolitres), t_{s1} is the starting time of line 1 operation (relative hour), t_{s2} is the starting time of line 2 operation (relative hour), t_{e1} is the ending time of line 1 operation (relative hour), t_{e2} is the ending time of line 2 operation (relative hour), R_{L1} is the rate at which line 1 fills the stock (hectolitres per hour), R_{L2} is the rate at which line 2 fills the stock (hectolitres per hour), t_{ts} is the starting time of truck loading operation (relative hour), t_{te} is the ending time of truck loading operation (relative hour) and R_T is the rate at which the truck is loaded (hectolitres per hour).

In Simulink, the stock is represented by an integrator block (section 3.3.3). Its value, which represents the volume of the products, is changing depending on the rate, which is calculated by a sum block (section 3.3.4), which adds up all of the rates listed above. The integrator and the sum block are connected with a serial connection.

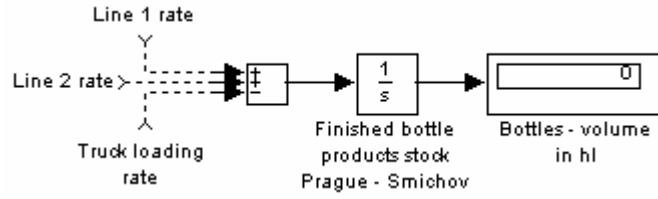


Fig. 4.16: Model of finished bottle products stock Prague – Smíchov

Analogically, the volume of barrel products stored in the second section of the stock, labeled as S_{SK} can be expressed mathematically as

$$S_{SK} = S_{SK0} + \int_{t_{sk}}^{t_{ek}} R_{L3} d\tau - \int_{t_{ts}}^{t_{te}} R_{TK} d\tau \quad (4.6)$$

where S_{SK0} is the initial volume of barrel products on the stock (expressed in hectolitres), t_{sk} is the starting time of line 3 operation (relative hour), t_{ek} is the ending time of line 3 operation (relative hour), R_{L3} is the rate at which line 3 fills the stock (hectolitres per hour), t_{ts} is the starting time of truck loading operation (relative hour), t_{te} is the ending time of truck loading operation (relative hour) and R_{TK} is the rate at which the truck is loaded (hectolitres per hour).

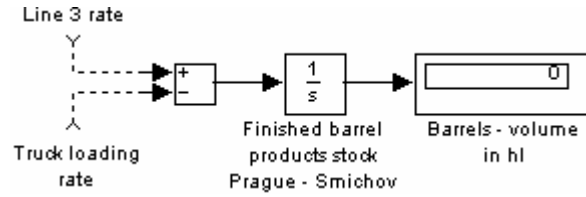


Fig. 4.17: Model of finished barrel products stock Prague – Smíchov

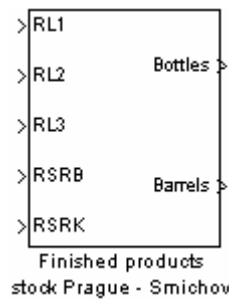


Fig. 4.18: Masked model of finished products stock Prague – Smíchov

4.2.3 Raw – material stock

Production lines 1 and 2 may use returnable containers in the production process described in section 4.2.1. For this purpose, there is a raw – material stock in the Prague –

Smíchov facility. Returnable containers, if needed, are moved to the production line feeder at the rate R_V which equals to the current production capacity of the line

$$R_V = C.E \quad (4.7)$$

The model must allow setting a time period, during which returnable containers will be used by one of the production lines and it must also provide the monitoring of the raw – material stock balance. This includes subtracting returnable containers used by production lines and adding new returnable containers delivered from central stock Prague – Radotín.

The principle of this stock is the same as described in section 4.2.2. Mathematically, the raw – material stock balance labeled as S_{SR} , can be expressed as

$$S_{SR} = S_{SR0} - \int_{t_{rs1}}^{t_{re1}} R_{V1} d\tau - \int_{t_{rs2}}^{t_{re2}} R_{V2} d\tau + \int_{t_{Trs}}^{t_{Tre}} R_{TR} d\tau \quad (4.8)$$

where S_{SR0} is the initial volume of returnable containers on stock, t_{rs1} is the starting time of line 1 using returnable containers (relative hour), t_{rs2} is the starting time of line 2 using returnable containers (relative hour), t_{re1} is the stopping time of line 1 using returnable containers (relative hour), t_{re2} is the stopping time of line 2 using returnable containers (relative hour), t_{Trs} is the starting time of truck unloading operation (relative hour), t_{Tre} is the ending time of truck unloading operation (relative hour) and R_{TR} is the rate at which the truck is unloaded (hectolitres per hour).

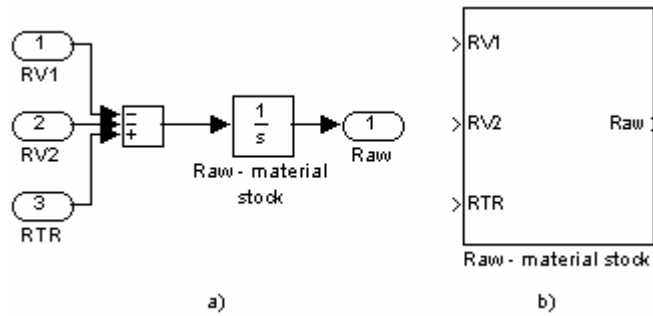


Fig. 4.19: a) model of raw – material stock, b) masked model of raw – material stock

4.3 Central stock Prague – Radotín

As it was suggested in section 2.2.1, the central stock facility in Prague – Radotín includes the following objects:

- Final products stock
- Raw – material stock

4.3.1 Final products stock

The Prague - Radotín finished product stock is divided into two sections. One stores bottle products and the other one stores barrel products that are delivered to Prague – Radotín from the brewery's production facilities. Products are stored there until they are transported with truck to secondary stocks or customers.

The volume of products stored in the first section of the stock, labeled as S_{RB} can be expressed mathematically as

$$S_{RB} = S_{RB0} + \int_{t_{ts}}^{t_{te}} R_{SRB} d\tau + \int_{t_{tos}}^{t_{toe}} R_{ORB} d\tau - \int_{t_{tcs}}^{t_{tce}} R_{ROB} d\tau \quad (4.9)$$

where S_{RB0} is the initial volume of bottle products on the stock (expressed in hectolitres), t_{ts} is the starting time of unloading a truck arriving from production facility in Prague - Smíchov (relative hour), t_{te} is the time when unloading of the truck arriving from production facility in Prague – Smíchov is finished, R_{SRB} is the rate at which the truck is unloaded (hectolitres per hour), t_{tos} is the starting time of unloading a truck arriving from other production facility (relative hour), t_{toe} is the time when unloading of the truck arriving from other production facility is finished, R_{ORB} is the rate at which the truck is unloaded (hectolitres per hour), t_{tcs} is the starting time of the truck transporting products to secondary stocks loading operation (relative hour), t_{tce} is the ending time of truck loading operation (relative hour) and R_{ROB} is the rate at which the truck is loaded (hectolitres per hour).

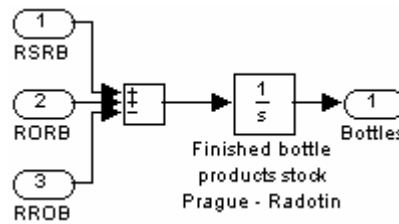


Fig. 4.20: Model of finished bottle products stock Prague – Radotín

The model of finished barrel products section can be described analogically

$$S_{RK} = S_{RK0} + \int_{t_{ts}}^{t_{te}} R_{SRK} d\tau + \int_{t_{tos}}^{t_{toe}} R_{ORK} d\tau - \int_{t_{tcs}}^{t_{tce}} R_{ROK} d\tau \quad (4.10)$$

where S_{RK0} is the initial volume of barrel products on the stock (expressed in hectolitres), t_{ts} is the starting time of unloading a truck arriving from production facility in Prague - Smíchov (relative hour), t_{te} is the time when unloading of the truck arriving from

production facility in Prague – Smíchov is finished, R_{SRK} is the rate at which the truck is unloaded (hectolitres per hour), t_{tos} is the starting time of unloading a truck arriving from other production facility (relative hour), t_{toe} is the time when unloading of the truck arriving from other production facility is finished, R_{ORK} is the rate at which the truck is unloaded (hectolitres per hour), t_{tcs} is the starting time of the truck transporting products to secondary stocks loading operation (relative hour), t_{tce} is the ending time of truck loading operation (relative hour) and R_{ROK} is the rate at which the truck is loaded (hectolitres per hour).

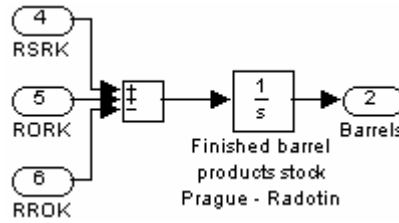


Fig. 4.21: Model of finished barrel products stock Prague – Radotín



Fig. 4.22: Masked model of finished products stock Prague – Radotín

4.3.2 Raw – material stock Prague – Radotín

The raw – material stock in Radotín stores returnable containers, which are collected from customers. These containers are then transported to the production facilities. As stated earlier, this work deals only with primary logistic and collecting returnable containers comes under secondary logistic, therefore this part is represented only by a subsystem, which ensures the correct adding and subtracting of the returnable containers, depending on if they are brought to or taken from the facility.

Analogically to the raw – material stock located in Prague – Smíchov (section 4.2.3), the volume of returnable containers labeled as S_{RRO} stored at this facility can be described mathematically as

$$S_{RR} = S_{RR0} + \int_{t_{TROs}}^{t_{TROe}} R_{ORV} d\tau - \int_{t_{TRs}}^{t_{TRe}} R_{RSV} d\tau - \int_{t_{TROOs}}^{t_{TROOe}} R_{ROV} d\tau \quad (4.11)$$

where S_{RR0} is the initial volume of returnable containers on stock (expressed in hectolitres), t_{TROs} is the starting time of unloading the truck arriving from secondary facilities (relative hour), t_{TROe} is the time when unloading of the truck arriving from secondary facilities is finished, R_{ORV} is the rate at which the truck is unloaded (hectolitres per hour), t_{TRs} is the starting time of loading a truck transporting returnable containers to Prague – Smíchov facility (relative hour), t_{TRe} is the time when loading of the truck transporting returnable containers to Prague – Smíchov facility is finished, R_{RSV} is the rate at which the truck is loaded (hectolitres per hour), t_{TROOs} is the starting time of the truck transporting returnable containers to secondary facilities loading operation (relative hour), t_{TROOe} is the ending time of truck loading operation (relative hour) and R_{ROV} is the rate at which the truck is loaded (hectolitres per hour).

In Simulink, the mathematical description of raw – material stock in Prague – Radotín can be represented as it is shown on the following figures.

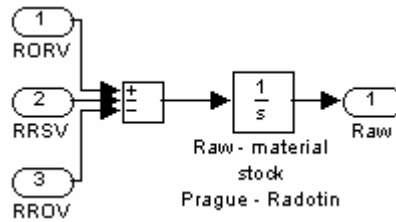


Fig. 4.23: Model of raw – material stock in Prague – Radotín

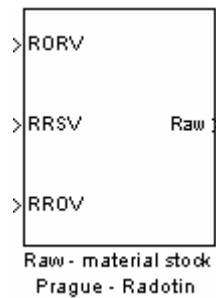


Fig. 4.24: Masked model of raw – material stock in Prague – Radotín

4.3.3 Finalized model

After designing all subsystems, it is possible to connect them with each other respecting all the relations and facts that were listed in previous sections. To avoid

confusing connections when using arrows, from and goto blocks were used (section 3.3.8). The model is split into three figures, showing the Prague – Smíchov facility, Prague – Radotín facility and transports between these and secondary facilities subsequently.

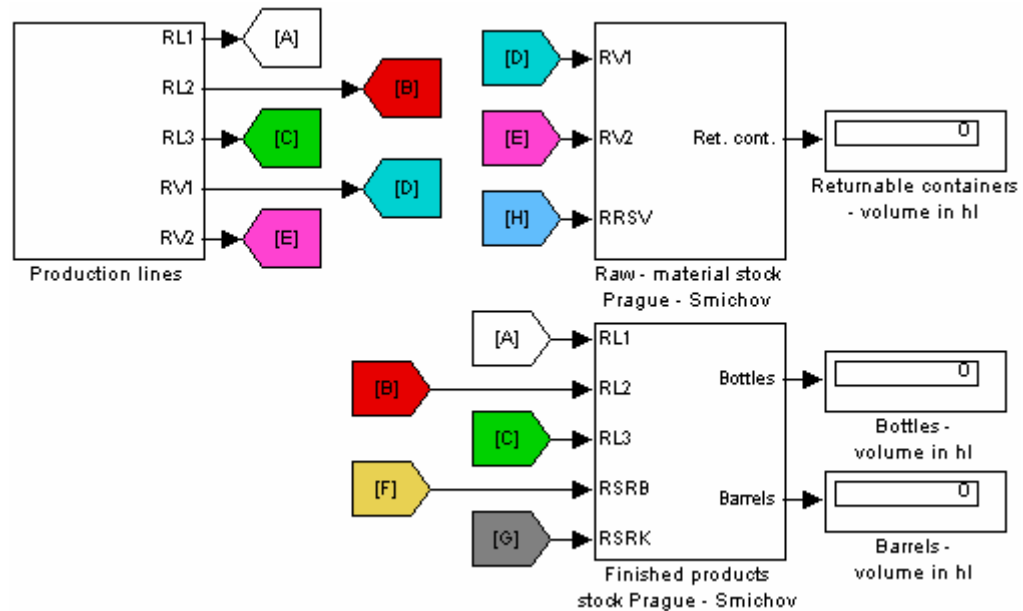


Fig. 4.25: Finalized model of Prague – Smíchov facility

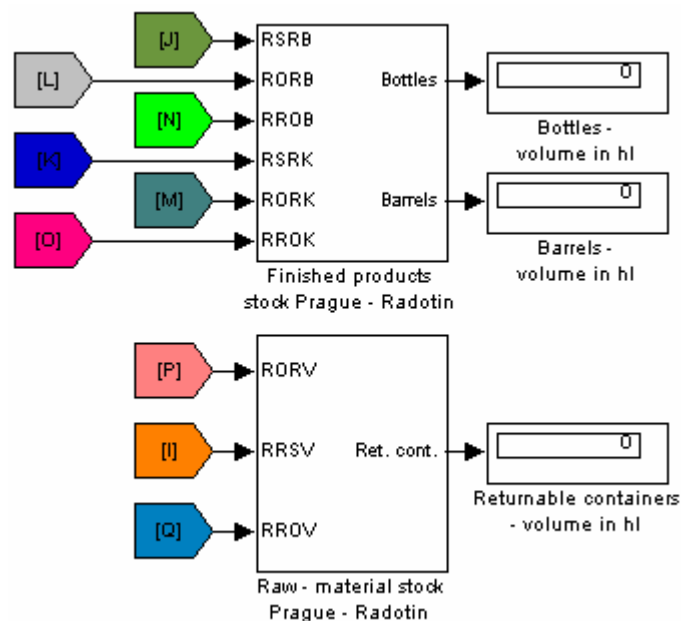


Fig. 4.26: Finalized model of Prague – Radotín facility

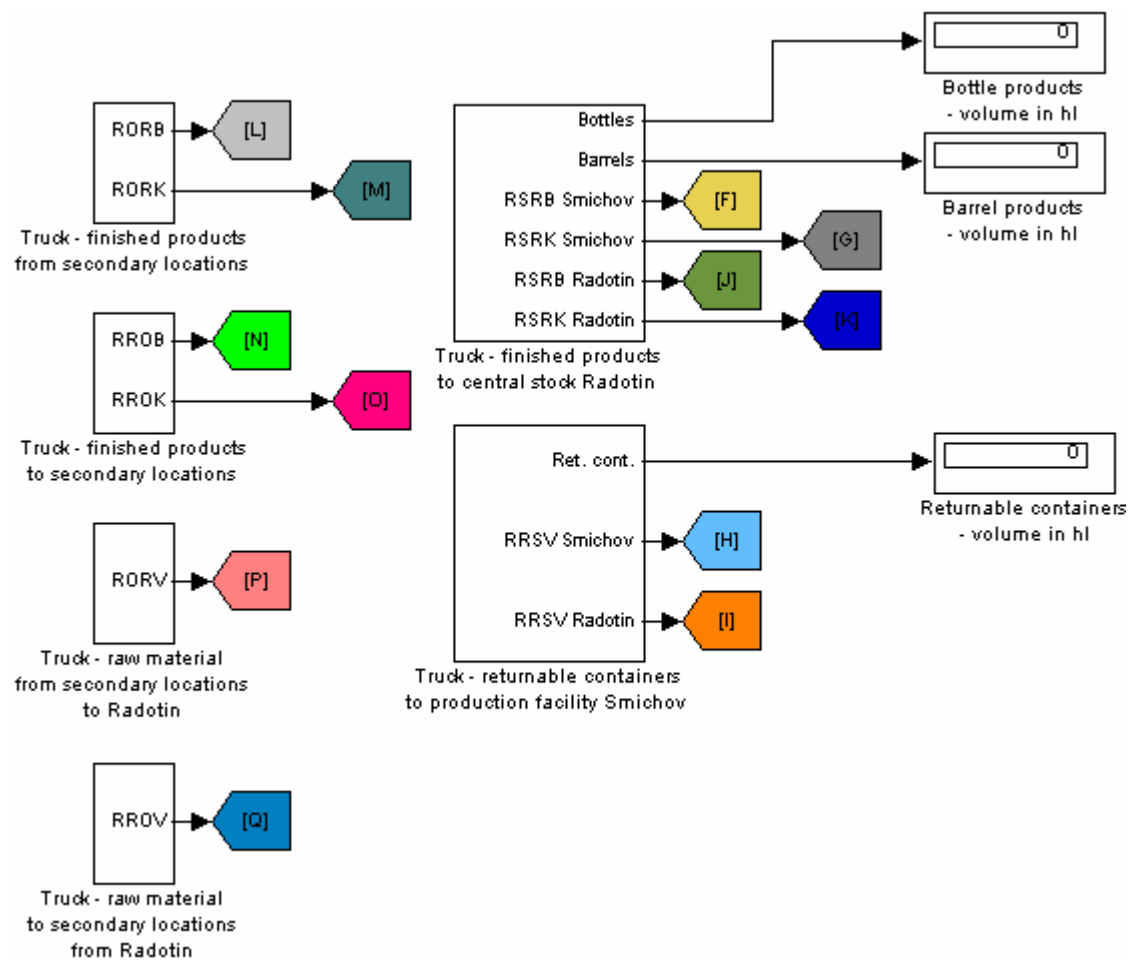


Fig. 4.27: Finalized model of logistic processes

Chapter 5

Model verification

After completing the construction of the model, it is needed to verify it, in order to find out if it is working as expected. This can be done by simulating all possible actions and comparing them with expected results.

5.1 Inputs and requested actions

Parameter values used for simulation were chosen in a way to represent as many different situations as possible. They are listed in the following tables.

Table 5.1: Input values – production lines in Prague – Smíchov

Parameter/Line	Line 1	Line 2	Line 3
Maximum capacity [containers/h]	24 000	24 000	300
Efficiency [%]	60	70	90
Waste rate [%]	0,1	0,1	0,1
Container volume [l]	0,5	0,33	40
Transport delay [h]	1,5	0,5	0,5
Start time [h]	0,5	1	0
Stop time [h]	6,5	7	6
Returnable containers usage start time [h]	0,5	1	-
Returnable containers usage stop time [h]	6,5	5	-

Table 5.2: Input values – stocks in Smíchov

Parameter/stock	Bottle products	Barrel products	Raw – material
Initial volume [hl]	100	50	800

Table 5.3: Input values – stocks in Prague – Radotín

Parameter/stock	Bottle products	Barrel products	Raw – material
Initial volume [hl]	8000	1000	2000

Table 5.4: Input values – transports between facilities

Parameter/route	S – R	R – S V	R – O	R – O V	O – R	O – R V
Start time [h]	4	2	5,5	6	5	3
Distance [h]	0,5	0,5	-	-	-	-
Bottle products [hl]	100	-	60	-	80	-
Barrel products [hl]	20	-	60	-	40	-
Raw-material [hl]	-	120	-	120	-	120

The route abbreviations mean the following:

- S – R: Smíchov – Radotín finished products transport
- R – S V: Radotín – Smíchov raw-material transport
- R – O: Radotín – secondary facilities finished products transport
- R – O V: Radotín – secondary facilities raw-material transport
- O – R: Secondary facilities – Radotín finished products transport
- O – R V: Secondary facilities – Radotín raw-material transport

5.2 Expected results of the model verification

The expected results of the simulation, which were calculated from the input parameter values listed in section 5.1, are shown in table 5.5.

Table 5.5: Expected results of the simulation

Stock item / Stock location	Prague – Smíchov	Prague – Radotín
Returnable containers [hl]	266	1 880
Bottle products [hl]	836	8 120
Barrel products [hl]	678	1 000

5.3 Results of the model verification

After running the model, with simulation time set to 10 hours, the results can be recorded in a table (table 5.6).

Table 5.6: Results of the model verification

Stock item / Stock location	Prague – Smíchov	Prague – Radotín
Returnable containers [hl]	266	1 883
Bottle products [hl]	836	8 119
Barrel products [hl]	679	1 000

The table shows, that the model is accurate. Slight differences are caused by different roundings of computer compared to the manual computing. Graphically, the results were as shown on the following figures.

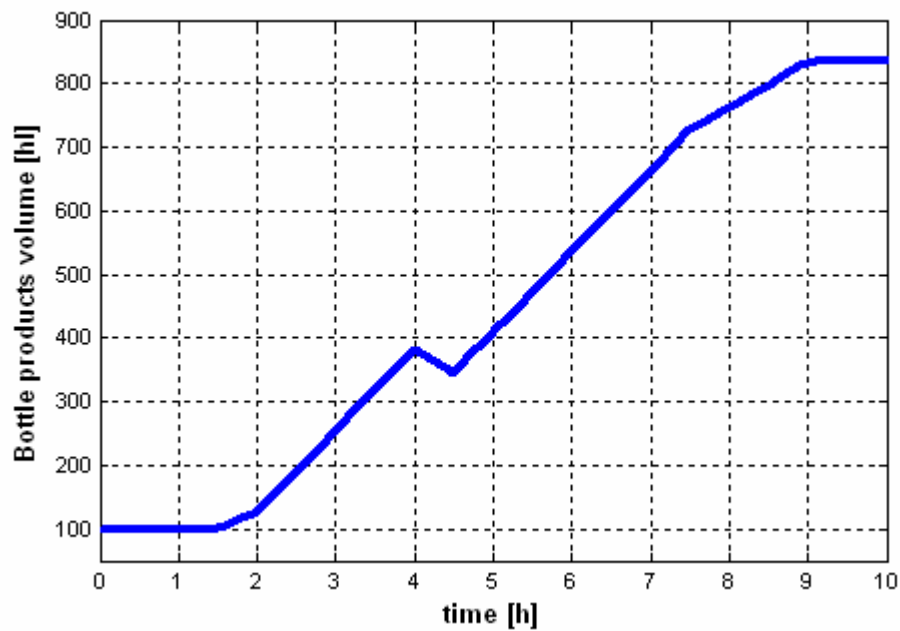


Fig. 5.1: Finished bottle products on stock Prague – Radotín

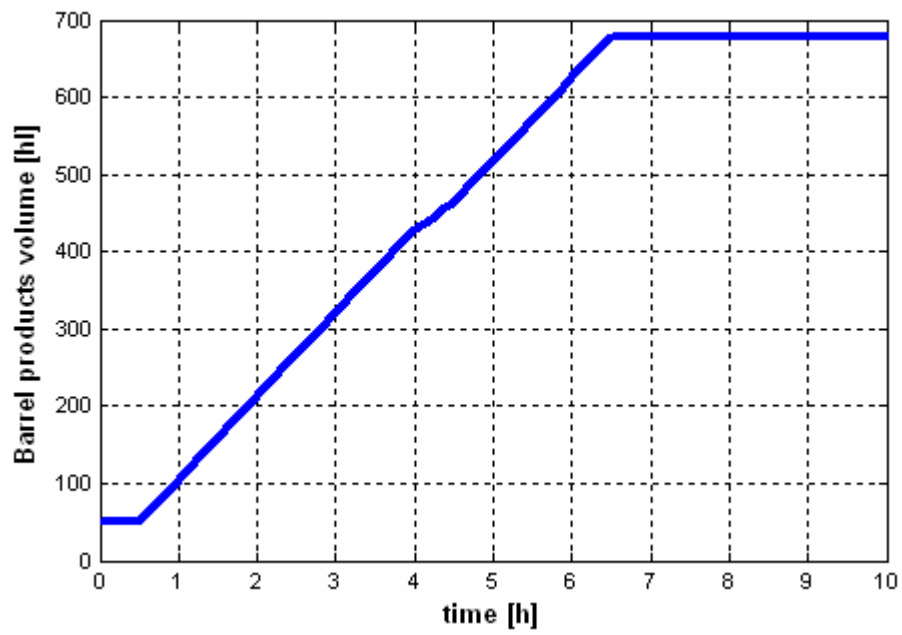


Fig. 5.2: Finished barrel products on stock Prague – Radotín

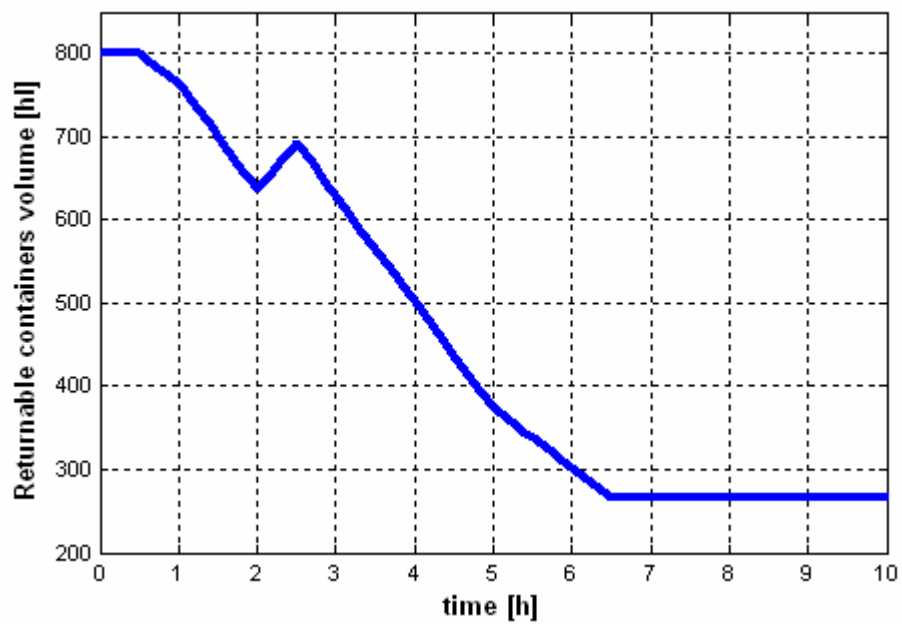


Fig. 5.3: Returnable containers on stock Prague – Radotín

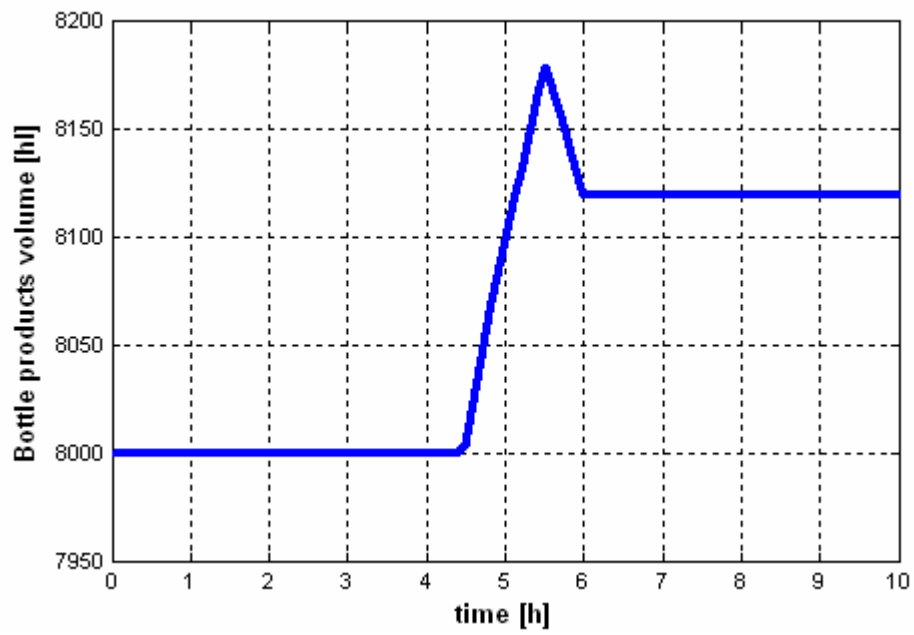


Fig. 5.4: Finished bottle products on central stock Prague – Radotín

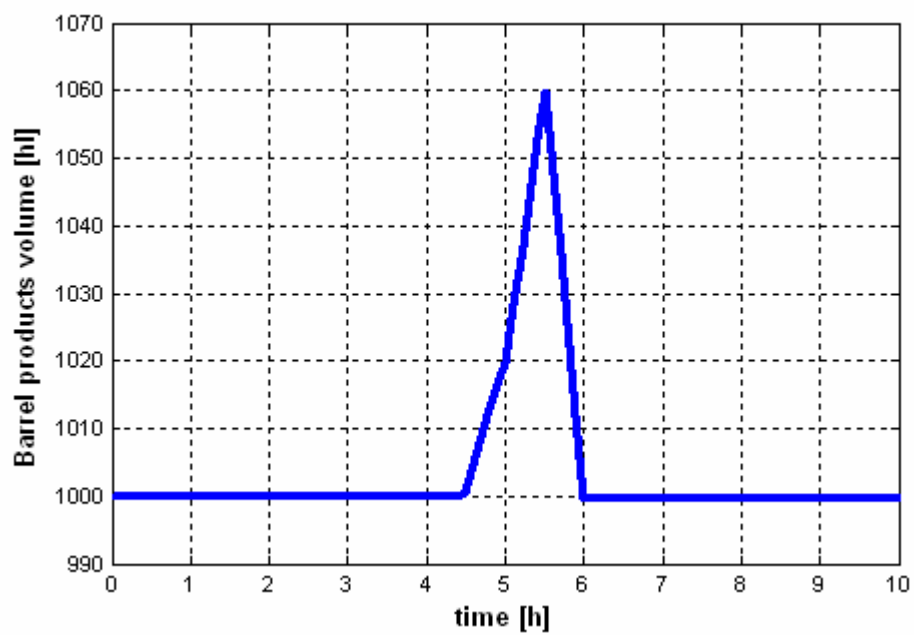


Fig. 5.5: Finished barrel products on central stock Prague – Radotín

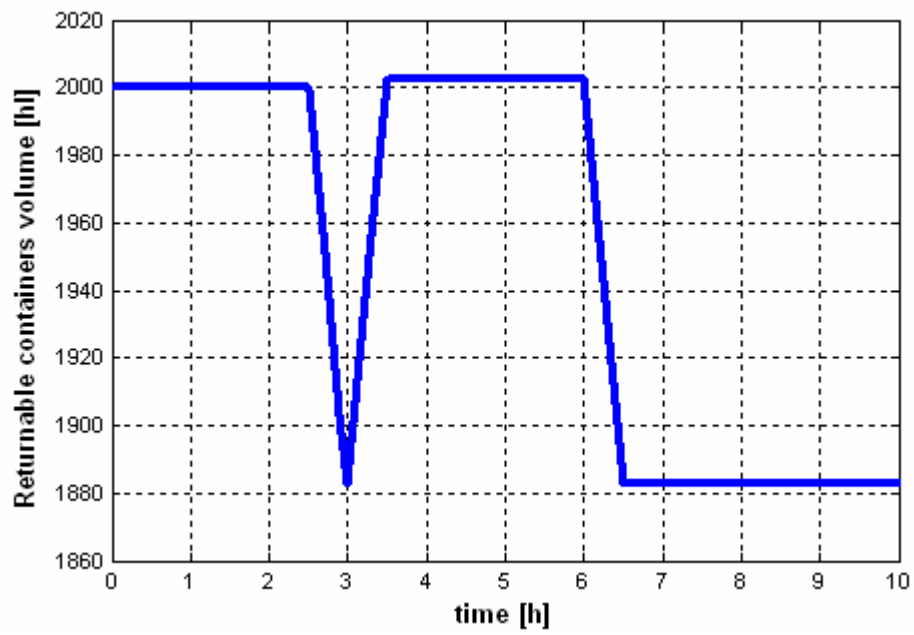


Fig. 5.6: Returnable containers on central stock Prague – Radotín

Chapter 6

Introduction to optimization

As it was already mentioned, this work is expected to be extended in the future, to enable not only simulation of the logistic processes, but also optimize them. At this time, model predictive control appears as the best way to solve the optimization of the logistic processes.

6.1 Optimization requests

The Staropramen brewery is a large corporation with many facilities, between which there are large material and product flows. Therefore the logistic planning department is constantly looking for tools, which could make their decisions more accurate and faster, thus making the corporation running more effectively.

The model with built-in optimization controller is expected to calculate the optimal material and product transfers between the brewery's facilities based on the values of production amounts in order to use the stocks and transportation the most effectively.

More detailed description of the optimization requests will be provided by the Staropramen brewery after the model is extended also to the secondary logistic simulation.

6.2 Model predictive control

The basic idea of the model predictive control technology is to find, based on the current system state and a defined reference for the controlled variable (system output) on

a prediction horizon, an optimal control sequence of the system input (manipulated variable) on the prediction horizon that minimizes an optimality criterion.

Then the optimal control sequence could be applied to the system on the prediction horizon, but the problem is that the prediction model can not ever be accurate, because disturbances of the system can not be predicted, the model of the system is only approximation of reality. Therefore only the first element of the found optimal control sequence is applied to the system and at the next time instance the optimization problem for new measured data is recalculated. This methodology is known as receding horizon control and brings feedback loop behaviour [4].

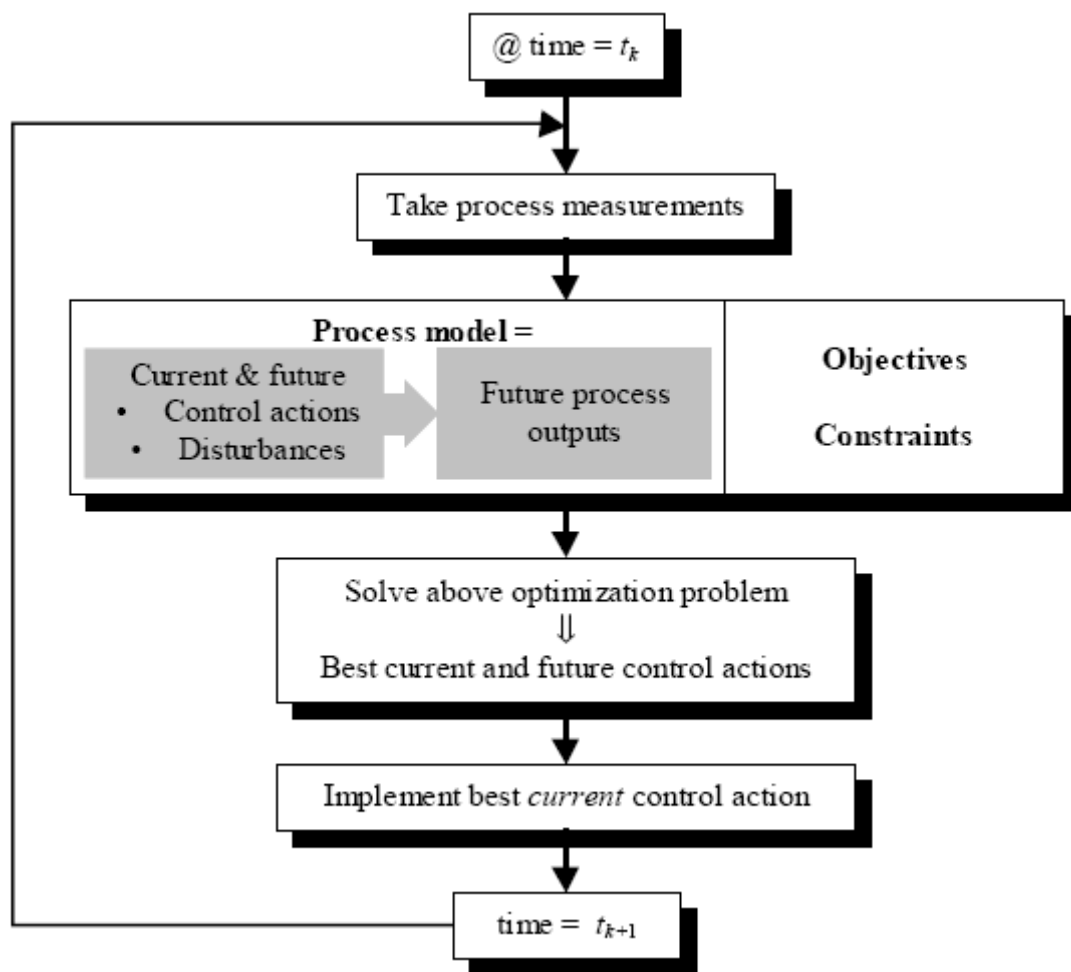


Fig. 6.1: Model predictive control scheme

Chapter 7

Final evaluation

This bachelor work's goal was to create a model representing the logistic processes in the Staropramen brewery. The results of the simulation, as documented in chapter 5, are corresponding with expected values.

Of course, not only the results are important. As already mentioned, this work is expected to be the base of another project, which is optimization of the brewery's logistic processes. This expectation had to be considered at each step of the model construction. The main problems, which I came across with when working on this project, were connected exactly with it. The final project handed to Staropramen brewery is going to be of great extent and it will use new evolving concepts (model predictive control) with which I lack experience. Therefore there is a possibility that, after receiving the first feedback from the brewery, some of the project areas will need to be redesigned in order to cover the needs of logistic – planning department more precisely. Also the controller, which will be designed in the following project, might need a different structure in some areas. This documents, how important the choice of structural approach to the model was.

In more detail, the problems related with the user inputs and their simulation. The solution, which is explained in the chapter dealing with model construction, is working with many fixed parameters which in real – world could vary in time. For example mechanic problems of trucks, traffic jams, problems with loading or unloading the cargo could result in longer transporting time. This is hard to simulate, although there are mathematical methods which can calculate the probability of these unexpected situations. But as this prediction is still not accurate, I do not expect the results to be much better as

using fixed values, since the logistic problems have to be solved immediately and it is not possible to predict them with that accuracy. If the testing phase will not show, that the current solution is accurate enough, I can see the only solution of this problem as an implementation of another user input, which would specify a logistic problem when it happens (for example traffic jam, expected delay is one hour) and letting the optimization controller recalculate the new conditions and suggest a solution.

Later, the logistic – planning department might want to keep track also of information, where is a specific product located. This feature could be added by dividing the stocks into more sections and specifying this section as the destination where the product is supposed to be stored. Also a slight adjusting of the production lines would be needed, in order to have the products divided from the beginning of the process. The basic principle of the model stays the same even after making these changes.

I believe that the rest of the model is representing the reality very accurately and provides a good starting ground for the following project which consists of adding the secondary logistic processes and an optimization controller to the model.

For the follow – up work, I have listed all the problems which I came across during the model construction and their solution in the appendix. This might serve as a reference and debugging aid.

References

- [1] Hyniová, K.: *Řídicí technika – Přednášky*, Vydavatelství ČVUT, Praha 2006
- [2] Horáček, P.: *Systémy a modely*, Vydavatelství ČVUT, Praha 2000
- [3] Bemporad, A., Morari, M.: *Control of systems integrating logic, dynamics, and constraints*, Automatica 35, 1999
- [4] Nikolaou, M.: *Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs*, Chemical Engineering Department, University of Houston
- [5] *The Mathworks* (<http://www.mathworks.com>)

Appendix A

List of problems and issues solved or in need to be solved:

- Automatic switching of the lines – currently, when performing maintenance or when the automated line is being prepared for a product change, it has to be defined manually. The main problem with this is, that the time delay does not have to be constant. Possible solution is to have an additional input, where user could define the time delay.
- Defining the state of the lines – currently, the lines are being controlled (on/off) by a rectangular signal with requested length. After testing it in real world simulations, it needs to be considered if this is the best solution.
- Truck loading and offloading – currently, all loading and unloading actions are set to an equal time, because the possible time differences are expected to be so small, that they will not affect the simulation. If the testing shows, that this idea is wrong, it must be corrected. Possible solution is suggested in final conclusion.
- Truck transports – currently, all transport routes are represented by one truck, but there are more trucks moving between the facilities. It does not effect the principle of the model, but at this state, it is redundant to implement more truck subsystems to the model. In the final model all the trucks must be implemented, in order to see the movement of each truck.
- Units – currently, products are measured in hectolitres. After testing phase, it needs to be reconsidered if also units in pallets must be added.
- Stocks – currently, only number of products on stocks is monitored. In the future, their distribution at the facility might be also considered.

Appendix B

Content of the attached CD:

- Bachelor work
- Matlab Simulink model of the logistic processes in the brewery