



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ENERGETICKÝ ÚSTAV

ENERGY INSTITUTE

**TEORIE FRONT A VIRTUÁLNÍ TOVÁRNA ZA POMOCI
MATEMATICKÉHO MODELU**

QUEUING THEORY AND VIRTUAL FACTORY BY USING MATHEMATICAL MODELS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Myška

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Mauder, Ph.D.

BRNO 2018

Zadání bakalářské práce

Ústav: Energetický ústav
Student: **Martin Myška**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. Tomáš Mauder, Ph.D.**
Akademický rok: 2017/18

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Teorie front a virtuální továrna za pomoci matematického modelu

Stručná charakteristika problematiky úkolu:

Simulace výrobního procesu pomocí matematických metod patří v dnešní době k nepostradatelným pomocníkům při plánování výroby, rozvržení servisních odstávek, k tvorbě nové výrobní linky, atd. I když se každý výrobce snaží o deterministický výrobní proces, tedy o přesně časově definované výrobní operace, ne vždy je tento požadavek plně zaručen. Mnohdy je tedy výrobní proces ovlivněn náhodnými vlivy. V některých případech se dá tato náhodnost zanedbat bez větších problémů, jindy by však její zanedbaní vedlo k nereálným výsledkům. Cílem práce je tedy ukázat na vlastním zjednodušeném modelu výrobní linky výsledné časy produkce pro plně deterministický proces stejně tak jako pro proces stochastický, při využití různých distribučních funkcí. Při zadávání výrobních časů by se mohlo rovněž navázat na probíhající spolupráci s firmou BOSCH DIESEL s.r.o. – Jihlava.

Cíle bakalářské práce:

Cílem bakalářské práce je popsání simulačního přístupu pomocí metod teorie front, vytvoření zjednodušené virtuální továrny simulující jednu výrobní linku v simulačním nástroji SIMULIK – SIMEVENTS včetně vlastní implementace zdrojového kódu v programu MATLAB. V práci bude rovněž rozebrán rozdíl mezi deterministicky a mezi stochasticky definovanými výrobními časy.

Seznam doporučené literatury:

GROSS, Donald, John F. SHORTLE, James M. THOMPSON a Carl M. HARRIS. Fundamentals of queueing theory. 4th ed. Hoboken: John Wiley & Sons, 2008. Wiley series in probability and statistics. ISBN 9780471791270.

RAO, Singiresu S. Engineering optimization: Theory and practice. Čtvrté vydání. Hoboken: Wiley, 2009. ISBN 978-0-470-18352-6.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18.

V Brně, dne 25. 10. 2017



doc. Ing. Jiří Pospíšil, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce se zabývá simulačním přístupem pro modelování výrobní linky, jejíž topologie vychází z reálné výrobní linky firmy BOSCH DIESEL s.r.o. – Jihlava. Problematika simulací je řešena v programu Matlab-Simulink-SimEvents. Představený model primárně slouží k pilotní simulaci výrobního procesu s možností plánování pravidelných odstávek strojů, nastavení aktivity pomocných pracovišť, volby stochastického a deterministického časového přístupu a hledání úzkých míst v současném výrobním řetězu. Jde o pilotní verzi simulačního modelu, a tedy oproti reálnému výrobnímu procesu pracuje s určitým zjednodušením, jako je např. zanedbání krátkých přestávek operátorů strojů apod. V práci jsou rozebrány výsledky a porovnání jednotlivých simulací, stejně jako možnosti zdokonalení představeného modelu, aby se více přiblížil reálnému výrobnímu procesu.

SUMMARY

This thesis deals with the modelling of a production line. The topology of the line is based on the real production line of BOSCH DIESEL s.r.o. – Jihlava company. Simulations have been made using Matlab-Simulink-SimEvents software. The created model represents a pilot simulation of the manufacturing process. The model allows to set a deterministic or stochastic approach and use of an auxiliary worker for planning machine maintenances and finding the bottlenecks in the production. As a pilot version the model is simplified, e.g. the short breaks of operators are neglected. The simulation results and comparisons are described in detail later in this document. In order to bring the simulation closer to the real manufacturing process, future improvement options are mentioned.

KLÍČOVÁ SLOVA

teorie front, chytrá továrna, simulační přístup, simulace diskrétních událostí

KEYWORDS

queueing theory, smart factory, simulation approach, discrete-event simulation

MYŠKA, M. *Teorie front a virtuální továrna za pomoci matematického modelu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 36 s. Vedoucí bakalářské práce Ing. Tomáš Mauder, Ph.D.

Prohlašuji, že jsem bakalářskou práci *Teorie front a virtuální továrna za pomoci matematického modelu* vypracoval samostatně pod vedením Ing. Tomáše Maudera, Ph.D. s použitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

25. května 2018

.....
Martin Myška

Rád bych poděkoval svému vedoucímu Ing. Tomášovi Mauderovi, Ph.D. za odborné vedení, vstřícné konzultace, cenné rady a celkovou pomoc při zpracování této bakalářské práce.

OBSAH

ÚVOD	1
1 ZÁKLADNÍ ÚVOD DO TEORIE FRONT	4
1.1 PŘÍCHOD ENTIT	4
1.2 POČET SERVERŮ A JEJICH NÁVAZNOST	5
1.3 DRUHY A KAPACITA FRONTY	7
2 ANALYTICKÝ PŘÍSTUP.....	8
2.1 LITTLEHO ROVNICE.....	8
3 SIMULAČNÍ NÁSTROJ	10
4 MODEL VÝROBNÍ LINKY	14
4.1 POPIS LINKY	14
4.2 MODEL	16
5 VÝSLEDKY SIMULACÍ	19
5.1 CASE1	19
5.2 CASE 2.....	23
5.3 CASE 3.....	26
5.4 CASE 4.....	28
5.5 SHRNUÍ.....	31
ZÁVĚR	33
SEZNAM POUŽITÝCH ZDROJŮ	34
SEZNAM PŘILOH.....	36

ÚVOD

Díky stále novému technologickému pokroku a růstu ekonomiky se svižným tempem zvyšují i nároky na produkci průmyslových továren. Pro dosažení požadované produktivity se do výrobních závodů implementují nové stroje, které jsou pro výrobu nezbytné. U těchto strojů je snaha o jejich co možná největší autonomii, a naopak je trendem omezit nutnost zásahu obsluhy během pracovního procesu. Zároveň stroje potenciálně představují velmi cenný zdroj informací přímo z výrobní linky. Data, která jsou schopna zaznamenat, mohou managementu poskytnout zásadní informace o aktuálním průběhu výroby i vyráběných produktech. Následně díky nim lze optimalizovat produktivitu celého závodu i zvýšit kvalitu výstupní komodity [1].

Získávání dat z jednotlivých výrobních stanovišť je dnes díky pokroku telekomunikačních sítí značně zjednodušeno. Ke slovu se zde dostává Internet věcí (Internet of Things, IoT) [2]. Pojem, který představuje spojení široké škály zařízení neboli věcí s internetem. Objekty připojené k internetu se mohou jednoznačně identifikovat pomocí unikátní číselné adresy, kdykoli mezi sebou komunikovat pomocí sdílených dat a k těmto kdykoli přistupovat. Tím mohou navzájem ovlivňovat své jednání a spolupracovat pro dosažení společných cílů. Koncept Internetu věcí je dnes spotřebitelům známý pro pojmy jako chytrá domácnost, chytrá města nebo autonomní vozy. Skrz tyto projekty na sebe IoT poutá značnou pozornost. Méně viditelným, avšak mnohem dynamičtěji rozvíjejícím se, je odvětví IoT v průmyslu. V industriální sféře je též možno se setkat s názvem průmyslový Internet věcí (Industrial Internet of Things, IIoT) [3], který označuje propojení hardware a software výrobních strojů s internetem. Spojením je možno ze senzorů získávat a shromažďovat cenné informace přímo z výrobní linky. Praxi sběru dat a jejich analýzu však firmy uplatňovaly pomocí svých vlastních metod již mnoho let. IoT má principiálně stejnou funkci a je tedy častou otázkou, zda poskytuje hodnotnější službu. Rozdíl oproti stávajícím rozhraním je především v rozsahu a principu možných operací a služeb, které jsou nabízeny. Kupříkladu v systému Internetu věcí mohou být v reálném čase analyzovány velké toky dat pomocí tzv. cloud computing¹ [1]. Navíc rozsáhlé množství dat může být uloženo v oddělených cloudových úložištích pro budoucí analýzu. Díky ní je možno dojít k informacím a dlouhodobým statistikám, kterých by dříve nebylo možné dosáhnout kvůli nedostatečně efektivnímu vlastnímu algoritmu a relativně malému úložnému prostoru. S těmito výsledky tak IoT poskytuje lepší vhled do operací v továrně, což ve finále může vést až k snížení provozních nákladů.

Právě díky nástupu Internetu věcí a jeho postupné integraci do výrobního prostředí je možný příchod Průmyslu 4.0 (Industry 4.0) [4]. Pojem pocházející z Německa a následně rozšířený do EU, Číny, Indie a dalších a zemí. Označení 4.0 referuje ke čtvrté průmyslové revoluci (po mechanizaci, elektrifikaci a IT). Víze je taková, že průmyslové korporace vytvoří globální internetové sítě, ke kterým připojí své stroje, továrny a sklady. Vzniknou tak kyberneticko-fyzické systémy [3] v podobě chytrých továren, chytrých strojů a chytrých skladů. Tyto představují systémy, které spouštějí a kontrolují samy sebe díky sdílení dat a jejich analýze inteligentním softwarem. S jeho pomocí budou schopny

¹ Poskytované serverové služby a aplikace dostupné přes síť

predikovat a reagovat na různé události s výrobním procesem spjatých. Pomocí inteligence objektů, schopnosti automatické diagnostiky a jejich samostatného učení, lze dosáhnout značného zlepšení průmyslové výroby jako celku.

Podstatnou myšlenkou je, že inteligentní chování by neměly vykazovat pouze výrobní stroje. Inteligencí musí disponovat i okolí strojů pro dosažení spolupráce všech objektů. Základem všeho bude chytrá továrna, která upraví způsob, jak je výroba koncipována. Toho se dosáhne kromě použití chytrých strojů také pomocí chytrých výrobků. Jednou z možností, jak zajistit inteligenci vyráběných produktů, jsou integrované zmenšené RFID štítky, pomocí kterých může být kus jasně identifikován a přesně lokalizován v průběhu výrobního procesu. Díky tomu bude výrobek přesně vědět, kde se nachází, jaký je jeho aktuální stav a kterými kroky výroby musí ještě projít pro dosažení požadovaného stavu. Na základě těchto znalostí tak může i samotný kus zvolit chod alternativní cestou pro své dokončení. Kupříkladu bude inteligentní výrobek schopen poskytnout pásovému dopravníku informaci o tom, do které větve výrobní linky má být transportován. Zároveň tak bude umožněna i efektivní výroba několika různých druhů výrobků současně, čímž se zlepší dynamika výroby a možnosti reakce na aktuální poptávku trhu. Skrz Internet věcí je v Průmyslu 4.0 řízena i logistika odchozích produktů a skladových zásob. V ideálním případě tak dosáhneme požadavků pro chytrou továrnu, která se řídí dle šesti základních návrhových principů Industry 4.0 [3, 5]. Jsou to:

- Interoperabilita, neboli spolupráce jednotlivých účastníků výrobního procesu díky možnosti společné komunikace skrz IoT.
- Virtualizace, tedy schopnost propojit virtuální model s reálným systémem.
- Decentralizace, která umožňuje autonomní rozhodování bez narušení společného cíle.
- Práce a odezva systému v reálném čase, což je základní podmínka pro rozhodování a řízení výroby v reálném čase.
- Orientace na služby přes IoT pro využívání služeb jako Cloud computing apod.
- Modularita, neboli rozložení systému na jednotlivé moduly řešící dílčí problémy.

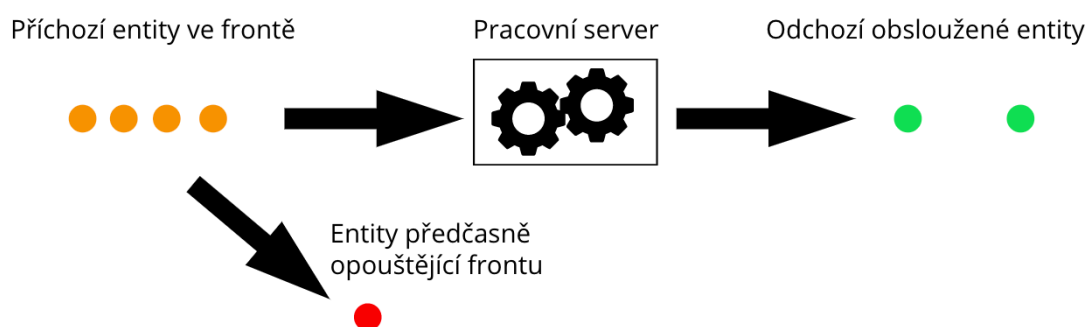
Jednou z možností programování již zmíněných inteligentních algoritmů užívaných pro automatickou diagnostiku a učení strojů je discrete-event simulation, což lze přeložit jako simulace diskretních událostí. Discrete-event simulation je počítačová simulace modelující zkoumaný reálný systém [6]. Vychází z jeho základních vlastností a aproximuje ty méně podstatné. Simulační technika vytváří model jako sekvenci diskretních událostí v čase. Každá událost namísto toho, aby probíhala kontinuálně v časovém úseku, se odehraje v konkrétně definovaný diskretní čas a vyvolá změnu v systému. Mezi dvěma po sobě jdoucími událostmi nedochází k žádné změně a je tedy možno okamžitě přecházet z jedné události na druhou [7]. Více o modelování pomocí discrete-event simulation lze najít v [8].

Discrete-event simulation je založena na pojmu teorie front. V české literatuře se také setkáme s označením teorie hromadné obsluhy [9]. Na jejím principu se navrhuje a řídí modely výše zmíněných simulací. A právě teorií front a vytvářením modelu virtuální továrny se zabývá tato práce. Nejprve je popsán základní princip a smysl teorie front. Na něj navazuje analýza charakteristik jednotlivých frontových systému a nastínění analytických řešení. Dále je popsán simulační přístup pomocí programu SimEvents. Jsou roze-

brány jeho základní funkce, možnosti nastavení a výstupy. Následně jsou poznatky z fungování programu uplatněny v praktické části, která popisuje model reálné výrobní linky. Je rozebráno a popsáno její zapojení. Dále jsou uvedeny a zhodnoceny poznatky nabyté při testování modelu v průběhu několika simulací. V závěru práce jsou zhodnoceny výhody a nevýhody tvorby simulačního modelu v programu SimEvents. Je zmíněna oblast využití a možnosti dalšího rozšíření představeného modelu. Zároveň je shrnuta celková aktuálnost tématu.

1 ZÁKLADNÍ ÚVOD DO TEORIE FRONT

Teorie front lze popsat na příkladu zákazníků, kteří mají zájem o jistou službu. Pokud však není služba dostupná okamžitě, musejí čekat ve frontě. V té mohou vyčkat na vyžadovanou službu, nebo řadu opustit a odejít bez servisu. Pokud v řadě vydrží, jsou obsluženi a místo opouštějí. Tento princip lze uplatnit jak pro lidskou společnost například při čekání na úřadě, čekání na výtah, v restauraci nebo u obchodní pokladny, tak pro jakékoli jiné objekty. Takové objekty nazýváme entitou [10], což je obecná jednotka našeho zájmu. Můžeme si pod ní představit letadla čekající na odlet nebo počítačový výpočet čekající na provedení. V této práci bude entita představovat součástku nebo výrobek. Typický příklad systému hromadné obsluhy ilustruje obrázek 1.1.



Obrázek 1.1: Schéma systému hromadné obsluhy [10]

Servis či službu zde bude vykonávat stroj neboli server, který pracuje daný procesní (výrobní) čas, což je čas strávený při provádění činnosti na entitě. Server bude označován písmenem S . Při spojení strojů za sebou vytvoříme výrobní linku. Ta se dělí na tři základní části a to vstup, výrobní proces a výstup [11].

Ve většině případů rozlišujeme základní charakteristiky, pomocí kterých můžeme popsat daný systém. Charakteristiky jsou nezbytné pro rozhodnutí, jakým typem matematického modelu lze celý systém nahradit [10]. Tyto charakteristiky jsou:

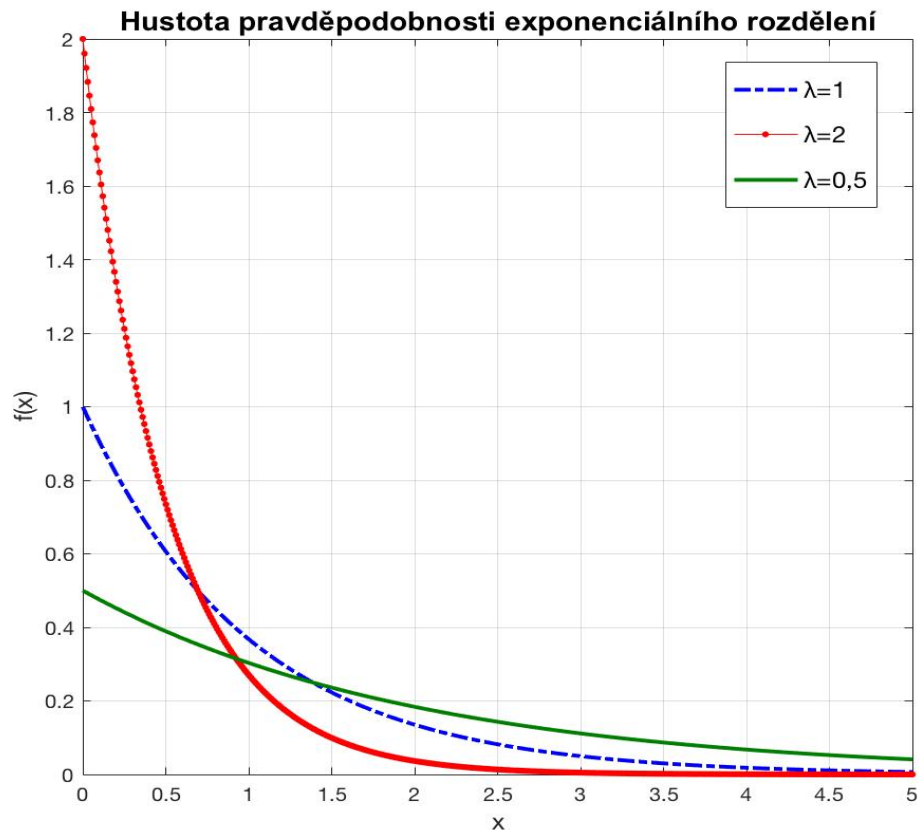
1. Vzorec příchodu entit
2. Počet a návaznost serverů
3. Charakteristika druhu a kapacity fronty

1.1 Příchod entit

Příchod jednotlivých entit může být modelován deterministicky, kdy za přesně stanovený časový úsek vstoupí entita do systému. Jedná se o ideální případ, u kterého lze přesně plánovat výrobu a řada výrobců se ho snaží dosáhnout. V reálném případě však bývá příchod dán určitou pravděpodobností tohoto jevu v daném časovém úseku a řídí se podle specifické distribuční funkce. Tento děj, kdy se čas příchodu řídí podle hustoty pravděpodobnosti, se nazývá stochastický [12]. Je nutno užít poznatků z pravděpodobnosti rozdělení pro popis časových intervalů mezi jednotlivými příchozími entitami. Zároveň je nezbytné vědět, zda entita může dorazit pouze samostatně, nebo i ve várce

s dalšími a pokud ano, znalost hustoty pravděpodobnosti popisující velikost příchozí skupiny je klíčová. V průběhu času se může vzorec charakterizující příchody entit měnit. V takovém případě hovoříme o nestacionární příchozí charakteristice. Pokud se distribuce pravděpodobnosti popisující vstup s časem nemění, máme charakteristiku stacionární [10]. Při simulacích se často užívá příchod entit podle exponenciálního rozdělení. Toto rozdělení se řídí dle rovnice (1.1) [13] a jeho vizualizace pro různé hodnoty parametru λ , který bývá interpretován jako intenzita vstupního toku [14], je na obrázku 1.2.

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & ; x \geq 0 \\ 0 & ; x < 0 \end{cases} \quad (1.1)$$



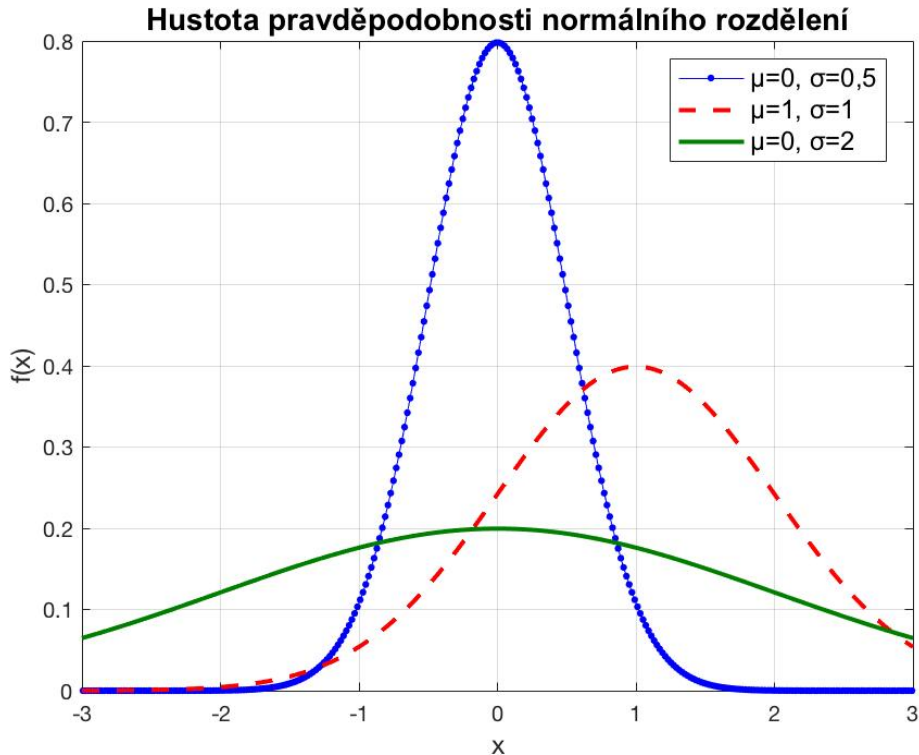
Obrázek 1.2: Hustota pravděpodobnosti exponenciálního rozdělení

1.2 Počet serverů a jejich návaznost

Základní charakteristikou serveru je jeho strojní neboli procesní čas, tedy doba, kterou stráví na provedení činnosti na entitě. Strojní čas rovněž může být modelován jako deterministický či stochastický. Pro náhodnou definici procesního času se často užívá Gaussovo (normální) rozdělení charakterizováno rovnicí (1.2) [13].

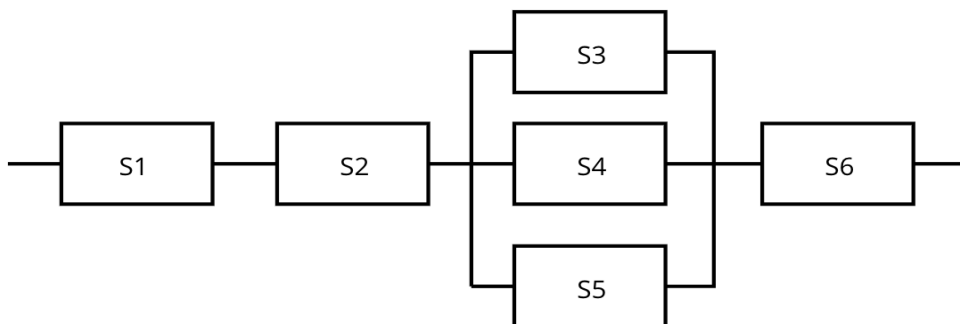
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], x \in (-\infty, \infty), \quad (1.2)$$

Platí, že μ , σ^2 jsou reálná čísla a $\sigma^2 > 0$. Na obrázku 1.3 je vizualizace hustoty pravděpodobnosti normálního rozdělení pro různé parametry μ a σ , kde μ je střední hodnota procesního času a σ označuje směrodatnou odchylku.



Obrázek 1.3: Hustota pravděpodobnosti normálního rozdělení

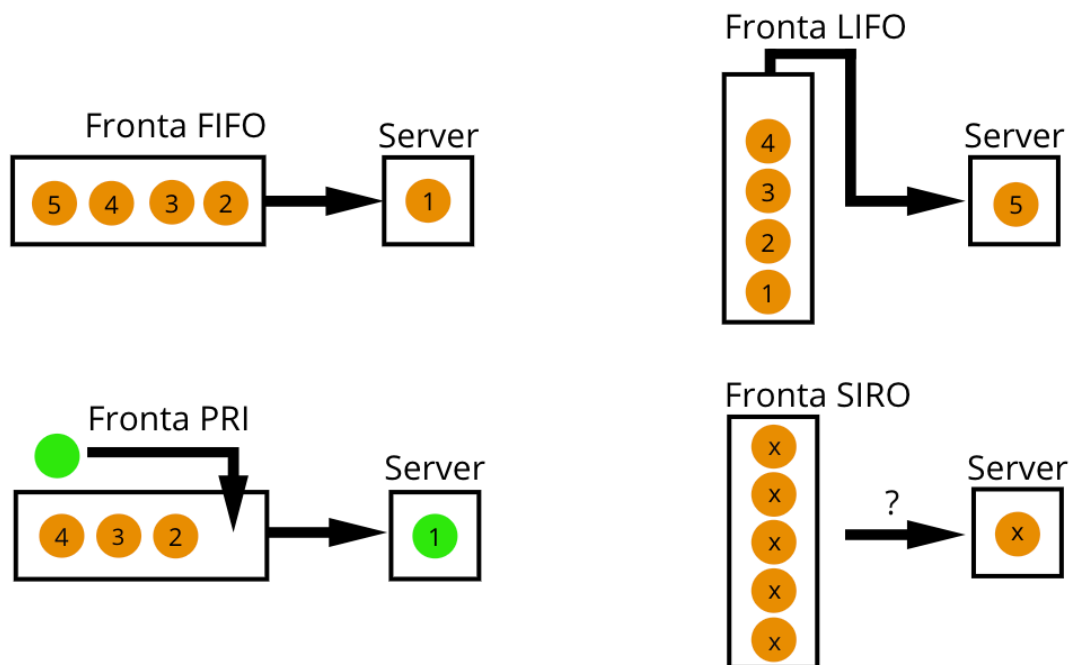
Další charakteristikou serveru je počet entit, které zvládne obsluhovat najednou. Standardně Entita během procesu nemusí procházet pouze jedním serverem, ale i vícero, což může zastupovat jednotlivé fáze výroby apod. Podle uspořádání rozlišujeme dva základní druhy zapojení serverů. Prvním typem je zapojení sériové. Stroje jsou zapojeny za sebou. Entita vstupuje na vstupu do prvního stroje a následně musí projít všemi servery v pevně daném pořadí podle sestavení strojů. Na obrázku 1.4 jsou takto zapojeny stroje *S1* a *S2*. Druhým způsobem zapojení strojů je zapojení paralelní neboli vedle sebe. Při tomto způsobu sestavení entita prochází právě jedním z *c* paralelně zapojených strojů. Spojením serverů paralelně dosáhneme toho, že můžeme vykonávat stejnou operaci na více pozicích. Tím se především v sériové výrobě šetří čas. Druhou pohnutkou může být nutnost vykonávání odlišných operací na různých entitách v jinak stejném výrobním procesu. Příkladem paralelně zapojených strojů na obrázku 1.4 jsou servery *S3*, *S4* a *S5*. Velmi často se uplatňuje kombinace zapojení strojů paralelně i sériově. Toto zapojení označujeme jako obecné [11]. Viz obrázek 1.4.



Obrázek 1.4: Obecné zapojení serverů

1.3 Druhy a kapacita fronty

V případě, že procesní čas serveru je větší než časový úsek, během kterého přijde další entita anebo přijde s požadavkem na server více entit ve stejný čas, vzniká fronta. Nejdůležitější charakteristikou fronty je její kapacita. V některých systémech je určitá fyzická limitace prostoru určeného pro čekání (velikost místnosti před přepážkou pošty). Při dosažení určitého, předem definovaného, počtu entit v řadě, dojde k zablokování vstupu a žádné další entity nejsou vpuštěny, dokud se místo ve frontě díky odchodu entity z ní neuvolní. Říkáme, že délka fronty je omezená. Opakem je fronta neomezená (nebo v realitě velmi velká), kde není žádná limitace počtu entit čekajících v řadě. Příkladem takovéto řady mohou být žádosti o výpůjčku nedostupné knihy z knihovny. Pod pojmem kapacita systému je myšlen součet kapacity fronty a počet entit který, zpracovává server v jeden časový okamžik. Intuitivní představa fronty je pravděpodobně ta, kterou známe z obchdů apod. Jedná se řadu typu FIFO (z anglického first in, first out). To znamená, že kdo přišel k obsluze první je také jako první obsloužen. Typicky se entity řadí za sebou. Dalším druhem řady je typ LIFO (last in, first out). Poslední entita, která byla do řady přidána, ji opouští jako první. S touto řadou se typicky můžeme setkat ve skladech, kde se na sebe vrší krabice. U fronty SIRO (selection in random order) vystupují z řady v náhodném pořadí. Také je možno se setkat s frontou PRI (priority) s odchodem podle priority požadavků [9, 15]. Rozdíly mezi druhy front jsou znázorněny na obrázku 1.5. Zároveň je nutné rozlišovat rozdíly mezi řadami, kdy se zákazníci ve frontě mohou svobodně rozhodnout frontu opustit, nebo entity vyčkají jakýkoli nutný čas pro získání obsluhy. Existují také případy, kdy entity mohou přecházet mezi jednotlivými frontami. Takovýto případ je možný u paralelně zapojených serverů, kde má každý svou individuální frontu [10].



Obrázek 1.5: Druhy front

2 ANALYTICKÝ PŘÍSTUP

Jeden ze způsobů řešení problematiky teorie front je analytický přístup. Na příkladu jednoduchých systémů budou uvedeny základní vztahy a výsledky užívané při vyvíjení modelů. Označme průměrný počet vstupujících entit do systému za časovou jednotku (intenzita vstupního toku) jako λ a průměrný počet obslužených entit za časovou jednotku (intenzita obsluhy) jako ν . Intenzita provozu c serverů je značena jako ρ [10].

$$\rho = \frac{\lambda}{c \cdot \nu} \quad (2.1)$$

Pokud $\rho > 1$, tedy $\lambda > c\nu$, průměrný počet entit vstupujících do systému převyšuje maximální průměrný počet obsluhovaných entit. Můžeme očekávat, že s plynoucím časem se fronta bude zvětšovat a za předpokladu, že bude volně umožněn vstup novým entitám do systému, poroste nade všechny meze. V tomto případě nikdy nedojde k ustálení ani stavu, kdy by server přestal pracovat. V případě $\rho = 1$ k ustálení rovněž nedojde, neboť náhodnost příchodu a procesního času zpracování entit znemožní, aby řada kdykoli zanikla. To by bylo možné pouze v případě deterministicky nastaveného modelu s přesným příchozím i procesním časem. Entity po příchodu nečekají v řadě a server je využíván nepřetržitě. V reálných podmínkách se stochastickým rozdělením však z toho vyplývá, že pro získání ustáleného stavu musí být vztah $\lambda/c\nu$ ostře menší než 1. Při znalosti intenzity vstupního toku entit a průměrné procesní rychlosti lze určit minimální počet paralelních serverů pro udržení $\rho < 1$ [10, 16].

2.1 Littleho rovnice

Rovnice (2.2) a (2.3) [10] představené panem Johnem D. C. Littlem jsou jedny z fundamentálních v celé teorii front. Dávají do spojitosti intenzitu vstupního toku a střední dobu strávenou ve frontě W_q (v systému W) se střední hodnotu počtu entit ve frontě L_q (v systému L). Platí pro jakýkoliv systém beze ztrát a o vstupním toku se předpokládá, že je homogenní.

$$L_q = \lambda \cdot W_q \quad (2.2)$$

$$L = \lambda \cdot W \quad (2.3)$$

Rovnice budou vysvětleny na následujícím příkladu. Příklad nemá poskytnout důkaz platnosti Littleho rovnic, ale ilustrovat jejich princip fungování a použití. Na obrázku 2.1 je znázorněn tok entit vstupujících do fronty systému v jednotlivé časové okamžiky. Uvažujeme, že přicházející entity se řadí do fronty dle systému FIFO a následně odcházejí do serveru, který v daný čas zpracovává vždy jen jednu entitu. Ze schématu vidíme, že za časový úsek $\langle 0; T \rangle$ vstoupily do fronty 3 entity. Střední počet entit ve frontě určíme výpočtem (2.4) s průměrnou dobu čekání entity v řadě spočtenou podle vztahu (2.5) [10].

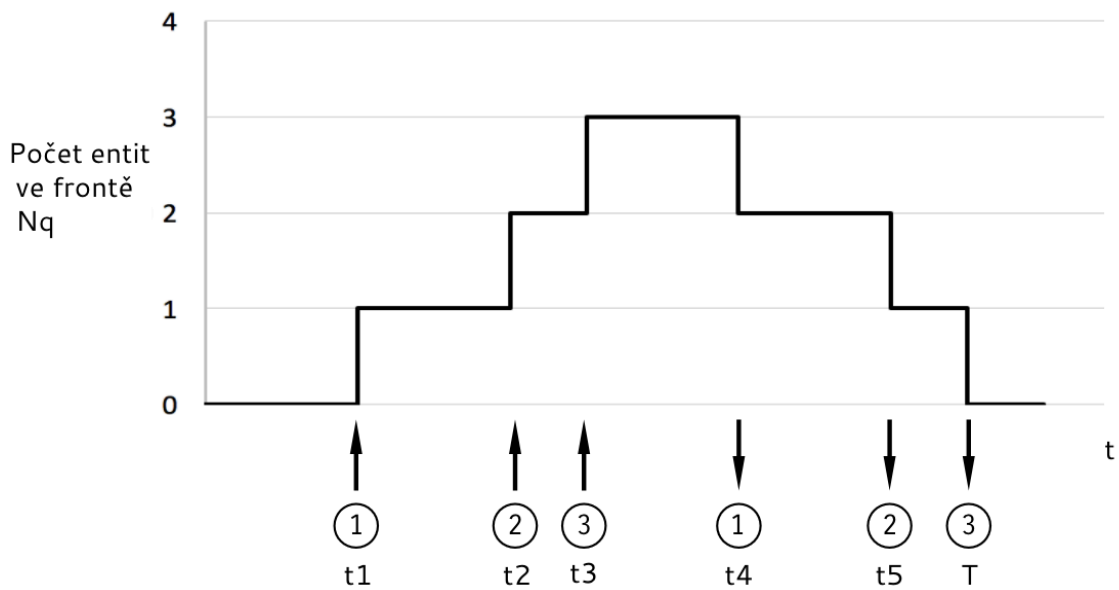
$$\begin{aligned} L_q &= \frac{[1 \cdot (t_2 - t_1) + 2 \cdot (t_3 - t_2) + 3 \cdot (t_4 - t_3) + 2 \cdot (t_5 - t_4) + (T - t_5)]}{T} \\ &= \frac{T + t_5 + 2 \cdot t_4 - t_3 - t_2 - t_1}{T} = \frac{\text{oblast pod křivkou}}{T} \end{aligned} \quad (2.4)$$

$$W_q = \frac{[(t_4 - t_1) + (t_5 - t_2) + (T - t_3)]}{3} = \frac{\text{oblast pod křivkou}}{N_q} \quad (2.5)$$

V rovnicích (2.4) a (2.5) vycházejí dva vztahy obsahující plochu pod křivkou. Po sloučení obou rovnic tedy dostaneme rovnici (2.6),

$$L_q = W_q \cdot \frac{N_q}{T} \quad (2.4)$$

ve které zlomek N_q/T vyjadřuje počet vstupujících entit za časový úsek, tedy se jedná o intenzitu vstupního toku λ . Dostáváme vztah $L_q = \lambda W_q$ což je zmíněná Littleho rovnice.



Obrázek 2.1: Entity ve frontě [10]

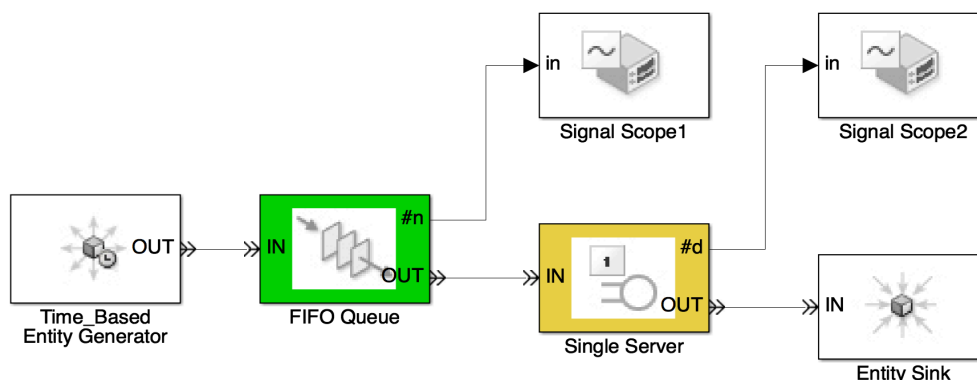
Analytické řešení teorie front je velmi užitečné k určení chování jednoduchého systému. Nicméně v průmyslové praxi, kde se výrobní závody sestávají z desítek či stovek strojů naráží analytika na limity svého užití. Příliš složité a komplexní systémy se dnes řeší pomocí simulačních technik. Právě tento způsob řešení je rozebrán v kapitole 3.

3 SIMULAČNÍ NÁSTROJ

Mimo analytického přístupu řešení teorie front nastíněného v kapitole 2 se k řešení problematiky teorie hromadné obsluhy používá přístup simulační. Ten pracuje na základě známých parametrů celého systému. Přitom zcela zásadním požadavkem je dodržení charakteristických rysů zadání. Výsledkem je model vytvořený ve vhodném software užívaný pro simulaci.

Programů pro vytvoření simulací existuje několik. Kupříkladu jsou to programy ARENA, SIMPROCESS nebo SIMUL8 [14]. Software použitý v této práci je SimEvents, což součást programu Matlab-Simulink. SimEvents poskytuje diskrétní simulační nástroj k vytvoření, řešení a analýzu modelů a optimalizaci jejich výkonu. Výkon lze zlepšovat kupříkladu zmenšením prodlevy mezi jednotlivými procesy, zlepšením průchodnosti systému nebo zamezením ztrátě paketů což může být definováno jako chyba při přenosu dat. Jeho součástí je knihovna komponent sdružující předdefinované bloky serverů, front, přepínačů atd. Díky tomu je umožněno modelovat chod jednotlivých entit, dopravního zpoždění, pracovního času a stanovení priorit pro dodržení rozvrhu. Pomocí SimEvents je možné studovat efekt časování úkolů a využití zázemí na produktivitu a chod systému. Tento nástroj je také vhodný pro hledání optimálního řešení při plánování odstávek a údržby strojů, předpovídání problematických míst ve výrobním procesu a návrhu struktury zásobování [17, 18].

Pro představení programu SimEvents využijeme jednoduchý model na obrázku 3.1. Vidíme zde základní konstrukční prvky užívané pro diskrétní modelování. Pod daným zapojením je možné si představit jednoduchý systém virtuální továrny s jedním strojem a vhodným nastavením parametrů lze tímto zapojením modelovat příklad užitý k demonstraci Littleho rovnic z kapitoly 2. Žádná simulace se nesestává pouze z bloků serverů, ale je potřeba do ní umístit i blok generátoru entit, blok pro řazení entit a blok jejich výstup ze systému. Entity procházejí modelem z generátoru entit dále dle logiky spojení označeného dvojitými šipkami. Spojení ukončené pomocí trojúhelníkových šipek neumožňuje chod entit, ale slouží k přenosu signálu. Ten následně ovlivňuje charakteristiku chování bloků v modelu. Funkce všech bloků bude nyní popsána.



Obrázek 3.1: Zapojení základních prvků

Time_Based Entity Generator je základním blokem fungujícím jako generátor entit v čase. Je to simulační náhrada příchodu entity do systému. Blok lze, stejně jako valnou většinu ostatních bloků v knihovně, dále nastavovat podle požadovaných parametrů.

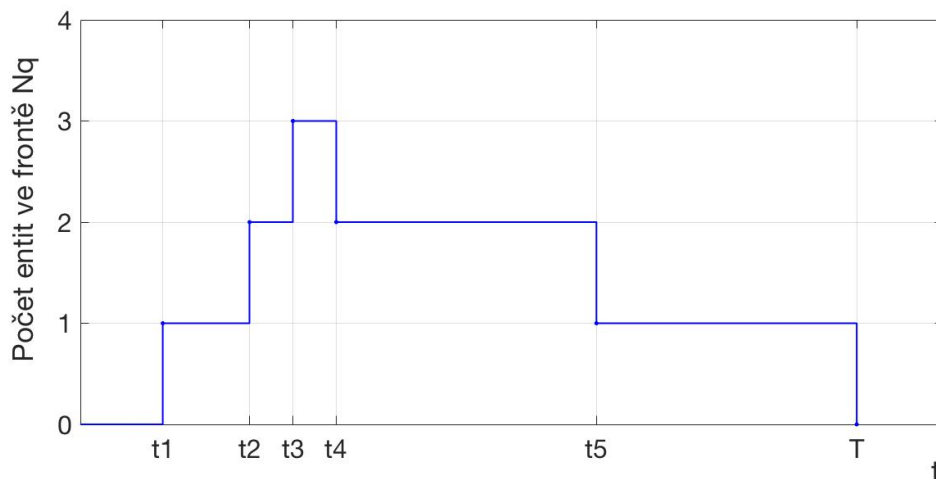
Parametry se nastaví v bloku před spuštěním simulace a tím se ovlivní jeho chování. Generování entit může být zvoleno jako konstantní v čase, kdy se vždy za stejnou periodu vyprodukuje jedna entita. Další možností je generování uniformní, kde ze zadaného minimálního a maximálního rozmezí jsou entity generovány náhodně dle rovnoměrného rozdělení. Poslední možností je náhodná generace entit s exponenciálním rozdělením. V bloku existuje kromě časově řízeného generování entit i možnost reakce generátoru na vstupní signál. Ten je možno posílat do bloku na základě jinak definovaného principu a tím řídit produkci entit.

Dalším v pořadí zapojeným blokem je FIFO Queue. Používá se v místech systému, se očekává vytvoření fronty. V modelech se jedná především o pozici před servery. Jediným charakterizujícím znakem fronty, která lze v bloku nastavit, je její kapacita. Po překročení této hodnoty se akce předcházející frontě zastaví. Blok umožňuje nastavení alternativního toku entit pomocí druhého výstupního portu. Tím odcházejí entity, které mají předdefinovanou maximální dobu čekání ve frontě a tuto hodnotu překročí. Software SimEvents, na rozdíl od konkurence, neposkytuje možnost později upravit pravidlo řazení entit v bloku [17]. Pro způsob řazení jiného než zobrazeného FIFO, je nutno vybrat z knihovny jiný typ řady. Program nabízí výběr mezi FIFO, LIFO a PRI.

Single Server představuje blok pro simulaci výrobního stroje, který na entitě provádí určitou operaci po dobu procesního času. Server pracuje vždy pouze na jedné entitě. Výrobní čas je možné předem definovat přímo v bloku pro všechny entity stejný, nebo se řídí podle atributu (vlastnosti) entity, anebo lze procesní čas řídit pomocí externího signálu jdoucího do Single Serveru. Stejně jako u bloků front, je možné i zde nastavit možnost alternativního odchodu entity z bloku. Tím lze například simulovat vyřazení vadné součástky ze stroje, která by server zablokovala. SimEvents knihovna současně nabízí servery typu Infinite a N. N_Server může být použit pro simulaci stroje, který dokáže zpracovávat více součástek najednou. Zároveň jej lze použít pro zjednodušení schématu vícero stejných, paralelně zapojených serverů. Je však nutné, aby všechny paralelní stroje měly stejný procesní čas, případně aby byl čas řízen atributem entity.

Základní pohnutkou každé simulace, je získávání dat používaných pro následnou analýzu. U většiny bloků lze aktivovat port pro odesílání vybraných informací. K získání a vizualizaci těchto dat slouží bloky Signal Scope, které se napojují na porty vybraných bloků. Po skončení simulace se zobrazí graf s vykreslenou charakteristikou. Na obrázku 3.2 je znázorněn vývoj počtu entit ve frontě FIFO Queue ze zapojení na obrázku 3.1 v závislosti na čase. Tento výstup z modelu v obecných časech odpovídá obrázku 2.1 z kapitoly Analytický přístup. Napojením dalšího bloku Signal Scope lze vykreslit charakteristiku L_q . Tímto je demonstrována ekvivalentnost simulačního přístupu vůči analytickému. Nabídka vypisovaných dat se u každého typu bloku mírně liší, nicméně mezi standardní možnosti zaznamenávaných informací patří:

- Počet odchozích entit
- Počet entit v bloku
- Průměrná délka čekání
- Průměrná délka fronty
- Počet entit předčasně opustivších blok
- Vytíženost serveru
- Čekající entity v serveru



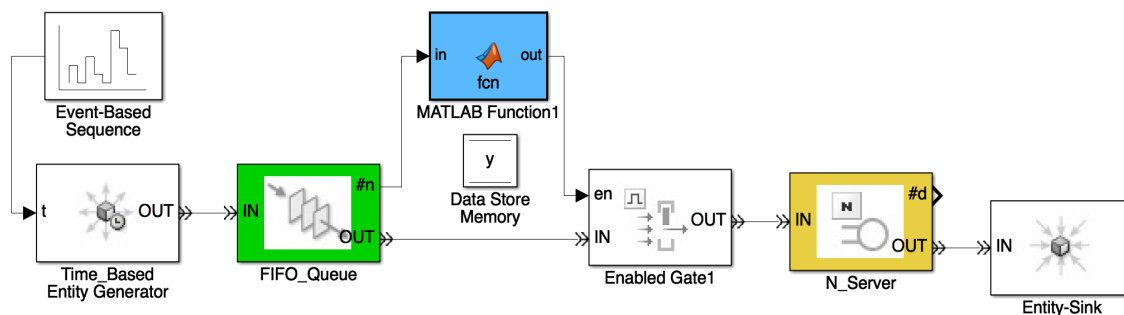
Obrázek 3.2: Entity ve frontě, simulace

Entity Sink blok používáme pro zakončení série bloků. Je to místo, ze kterého již entity v modelu dále nepokračují. V realitě si lze takto představit výrobky zabalené do krabice a připravené k expedici.

Toto jsou základní bloky, které jsou v následujícím modelu použity. SimEvents jich poskytuje celou škálu s množstvím různých nastavení. Navíc je mimo předdefinované bloky možné doprogramovat a implementovat do struktury vlastní zdrojové kódy z Matlabu. Těmi je možno řešit inteligenci a logiku systému při rozvrhování entit. Používají se zejména tam, kde předdefinované bloky SimEvents neposkytují dostatečně variabilní možnosti v sestavování modelu pro popis jeho charakteristických vlastností [11]. Jako příklad je na obrázku 3.3 ukázáno jednoduché zapojení. Oproti příkladu zapojení uvedeného na obrázku 3.1 se zde vyskytuje navíc blok Event-Based Sequence, kterým lze ovládat celkový počet vygenerovaných entit. Zapojením Enabled Gate1 do cesty toku entit je kontrolován jejich chod v systému. Blok je ovládán na základě binárních hodnot posílaných do vstupního portu. Na pozitivní signál brána reaguje otevřením a propouští entity dále do N_Server. V případě negativního signálu brána entity z FIFO_Queue nepouští dále a ty se ve frontě hromadí. Data Store Memory blok slouží pouze pro uchování globální proměnné. Samotný blok MATLAB Function1 ovládá otevírání a uzavírání brány. V tomto konkrétním případě zajišťuje poslání vždy pěti entit najednou pomocí jednoduchého kódu uvedeného v algoritmu 1.

Algoritmus 1: MATLAB Function1

```
function out = fcn(in)
    global y
    if in==5           %not sending until "in" entities in queue
        y=1
    elseif in==0
        y=0
    end
    out=y
end
```



Obrázek 3.3: Zapojení s MATLAB blokem

Před spuštěním každé simulace je nutné nastavit tzv. Simulation stop time, což je časový úsek, po kterém se simulace zastaví. Model tak simuluje chování systému ve zvoleném časovém intervalu. Samotné časové jednotky simulace jsou bezrozměrné. Je tedy čistě na uživateli, zda jednou časovou jednotkou prohlásí sekundu, minutu, hodinu či jakoukoli jinou jednotku času.

Mimo to je možné propojit bloky SimEvents s jinými z knihovny Simulinku. Tím lze rozšířit pole možného uplatnění simulace diskrétních událostí pomocí programu SimEvents do dalších oblastí. Více o programu SimEvents lze najít v [19]. Pro uplatnění vytvořených simulačních modelů v komerčním prostředí je zapotřebí zakoupit licenci Standard na programy Matlab, Simulink a SimEvents od společnosti MathWorks. Ceny těchto produktů před zdaněním jsou uvedeny v tabulce 3.1.

Tabulka 3.1: Ceník produktů [20]

	MATLAB	SIMULINK	SIMEVENTS
Roční licence	800 €	1200 €	1140 €
Trvalá licence	2000 €	3000 €	2850 €

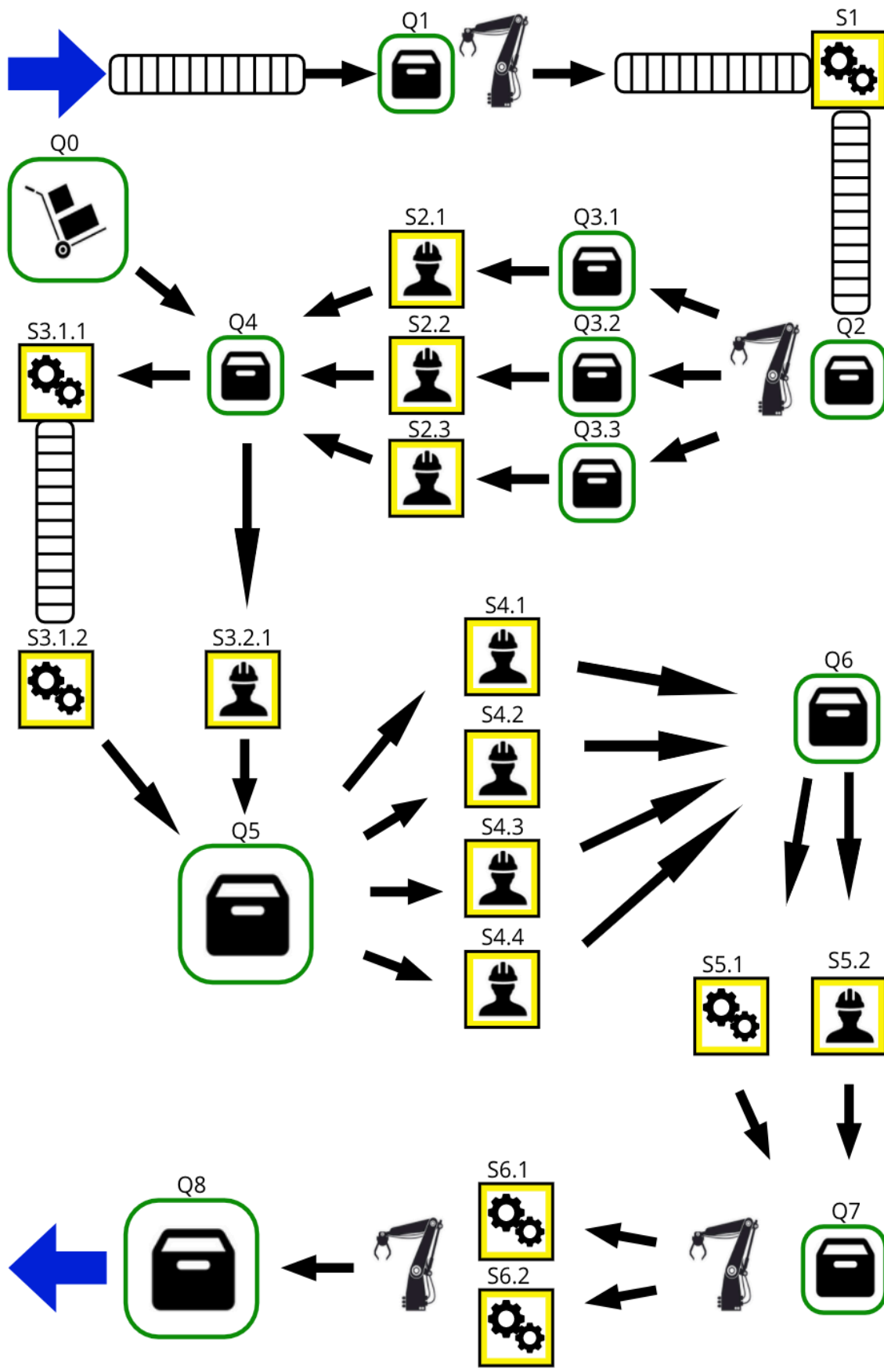
4 MODEL VÝROBNÍ LINKY

Na základě spolupráce s průmyslovou společností BOSCH DIESEL s.r.o. – Jihlava byl vytvořen simulační model reálné výrobní linky, která je součástí rozsáhlé výrobní haly. Výrobky jsou dodávány z linky ji předcházející a po průchodu pokračují k dalšímu zpracování. Pro vytvoření modelu byl použit simulační nástroj Simulink-SimEvents. Schéma topologie modelované linky je na obrázku 4.1. Při simulaci bylo nutné dodržet základní charakteristiky systému a zvolit vhodnou aproximaci tam, kde to bylo možné.

4.1 Popis linky

Základním ukazatelem proudu entit ve schématu 3.1 jsou modré šipky, které vyznačují jejich příchod a odchod ze systému. Logika přesunu výrobků v rámci samotné linky je značena černými šipkami. Pohyb kusů, jak schéma naznačuje, je zajištěn pásovými dopravníky nebo robotickými rameny. V případě, že není piktogramem specifikován způsob přesouvání entit, zajišťuje jejich pohyb obsluha. Zeleně orámované bloky s označením *Q* představují reálné sklady rozmístěné podél linky, kde se výrobky v průběhu výrobního procesu mohou hromadit. Žlutě označené bloky nadepsané písmenem *S* představují pracovní servery. Ty se dělí podle piktogramu na stroje bez obsluhy a na servery kde je nutná přítomnost operátora. Na všech těchto pozicích, kde je přítomna obsluha, bude při různých simulacích nastaven deterministický a stochastický výrobní čas. Cílem je porovnat vliv rozdílných nastavení procesních časů na celkové výsledky toku entit. Na pozicích, kde server nevyžaduje přítomnost operátora bude vždy zachován deterministicky nastavený výrobní čas. Zároveň je v modelu nutno podchytit možnost, že je server nečinný v předem přesně definovaném časovém úseku, což je například v době, kdy má operátor ze zákona nárok na přestávku.

Model linky popisuje chování systému, kde jako vstup jsou příchozí entity z vedlejší haly. Ty jsou přiváděny pomocí pásového dopravníku vždy po jednom kuse. Tento příchod entit bude později referován jako primární zdroj. Následně se kusy hromadí do beden ve skladu *Q1*. V okamžiku, kdy se nashromáždí pět entit v jedné krabici, přenesení je robotické rameno na pásový dopravník vedoucí k serveru *S1*. Tento stroj vždy zpracovává 5 kusů najednou. Jeho funkce nevyžaduje obsluhu, a proto je jeho výrobní čas vždy stanoven deterministicky. Po zpracování všech pěti výrobků se celá krabice přesune do skladu *Q2*. Robotickým ramenem je dále přenesena primárně do jedné z front *Q3.1*, *Q3.2* vždy tak, aby standardně pracovaly jen dva servery *S2.1* a *S2.2*. Pokud jsou sklady u těchto serverů zaplněny, rameno přesouvá výrobky do fronty *Q3.3*. Operátor stanoviště *S2.3* se zapojí pouze v případě nabytí počtu 25 entit ve frontě *Q3.3* a pozici opouští až ve chvíli úplného vyprázdnění tohoto skladu. Servery *S2.1*, *S2.2* a *S2.3* posílají entity do skladovacího prostoru *Q4*, který zaujímá specifickou pozici. Kusy jsou odtud primárně posílány do stroje *S3.1.1* vyznačujícího se tím, že zpracovává vždy 20 výrobků najednou. Před odesláním z *Q4* tak musí entity vyčkat, až se jich nashromáždí čtyři krabice po pěti. Pomocí pásového dopravníku jsou pak z pracoviště *S3.1.1* výrobky přemístěny do stroje *S3.1.2*, kde probíhá navazující část operace a odtud putují dále do skladu *Q5*. V případě, že stroj *S3.1.1* nestíhá kusy shromažďované na pozici *Q4* zpracovávat a jejich počet překročí 80, zapojuje se do výrobního procesu pomocný operátor na pozici *S3.2.1*. Ten zpracovává entity po pěti, avšak jeho výrobní čas je oproti stroji *S3.1.1* kratší. Navíc vykonává na své pracovní pozici i operaci stroje *S3.1.2*. Při deterministicky stanoveném procesním



Obrázek 4.1: Schéma výrobní linky

čase tak stanoviště *S3.2.1* zvyšuje propustnost pozic *S3* o 50 %. Entity jsou rovněž předávány do skladu *Q5*. Pokud je naopak na skladě *Q4* entit nedostatek a hrozilo by zastavení strojů *S3.1.1* a *S3.1.2*, je zajištěn přísun alternativní cestou ze skladu *Q0*, který symbolizuje příchod kusů z jiné části výrobní haly. Tento způsob doplnění kusů je určen pouze pro nárazové navýšení jejich počtu v cyklu linky a nikoli pro stabilní zásobování. V následujících částech bude na tento přísun entit odkazováno jako na sekundární zdroj. Z velkokapacitního skladu *Q5* jsou kusy distribuovány primárně na tři pozice *S4.1*, *S4.2* a *S4.3*. Požadavek na obsazení pozice operátora serveru *S4.4* přichází až ve chvíli, kdy paralelní servery nestíhají výrobky zpracovávat a množství entit ve frontě *Q5* dosahuje kritické hodnoty předem stanovené na počtu 60 kusů. Operátoři strojů *S4* pak hromadí výrobky ve skladu *Q6*, odkud jdou primárně do stroje *S5.1* nebo v případě nutnosti na pomocné stanoviště *S5.2*. To opět vyžaduje obsluhu operátora a je uvedeno do provozu pouze v případě překročení počtu 40 entit na pozici *Q6*. V momentě, kdy počet kusů ve skladu *Q6* klesne na 15, operátor pozice *S5.2* činnosti zanechá a dále je zpracovávání opět zprostředkováváno pouze strojem *S5.1*. Ze skladu *Q7* jsou pak výrobky již plně automaticky transportovány do serveru *S6.1* nebo *S6.2* a odtud dále do velkokapacitního skladu *Q8*. Z něj jsou entity odváženy z výrobní linky k dalšímu zpracování výrobního procesu. Všechny podpůrné stanoviště, tedy *S2.3*, *S3.2.1*, *S4.4*, *S5.2*, nepředpokládají stálý provoz. Je zde tedy předpoklad, že všechny tyto pozice zvládne obsluhovat jediný operátor. Žádné dva z podpůrných serverů tedy nemohou pracovat najednou. Způsob simulace této problematiky je dále rozebrán.

4.2 Model

V programu SimEvents byla na základě schématu linky na obrázku 4.1 navržena její simulace. Model, jehož zapojení je na obrázku 4.2, se z většiny sestává ze základních bloků, jejichž funkce je vysvětlena v kapitole 2. Stroje jsou simulovány jako *N_Server* bloky s počtem serverů nastavených podle množství entit, které zpracovávají najednou. Sklady pro hromadění entit jsou modelovány FIFO frontami. Navíc se zde vyskytují bloky *Clock* zaznamenávající čas v průběhu simulace. Dále bloky *Timed to Event Signal* pro převod časového signálu z *Clock* na event signal, se kterým je možné dále pracovat. Propojením těchto s *MATLAB Function* a *Enabled Gate* lze matlabovským kódem nastavit předem definovaný časový úsek, kdy server nezpracovává žádné entity. V případě deterministicky určených výrobních časů, jsou hodnoty stanoveny přesně v každém serveru. Pro modelování stochastiky slouží bloky *Event-Based Random Number*, které umožňují volbu distribuční funkce, na jejímž základě náhodně určují výrobní čas pro každou entitu. U strojů, kterých se to týká, tedy *S2.1*, *S2.2*, *S2.3*, *S3.2.1*, *S4.1*, *S4.2*, *S4.3*, *S4.4* a *S5.2*, je užito normálního rozdělení. Generování entit je podřízeno náhodnosti na základě exponenciálního rozdělení.

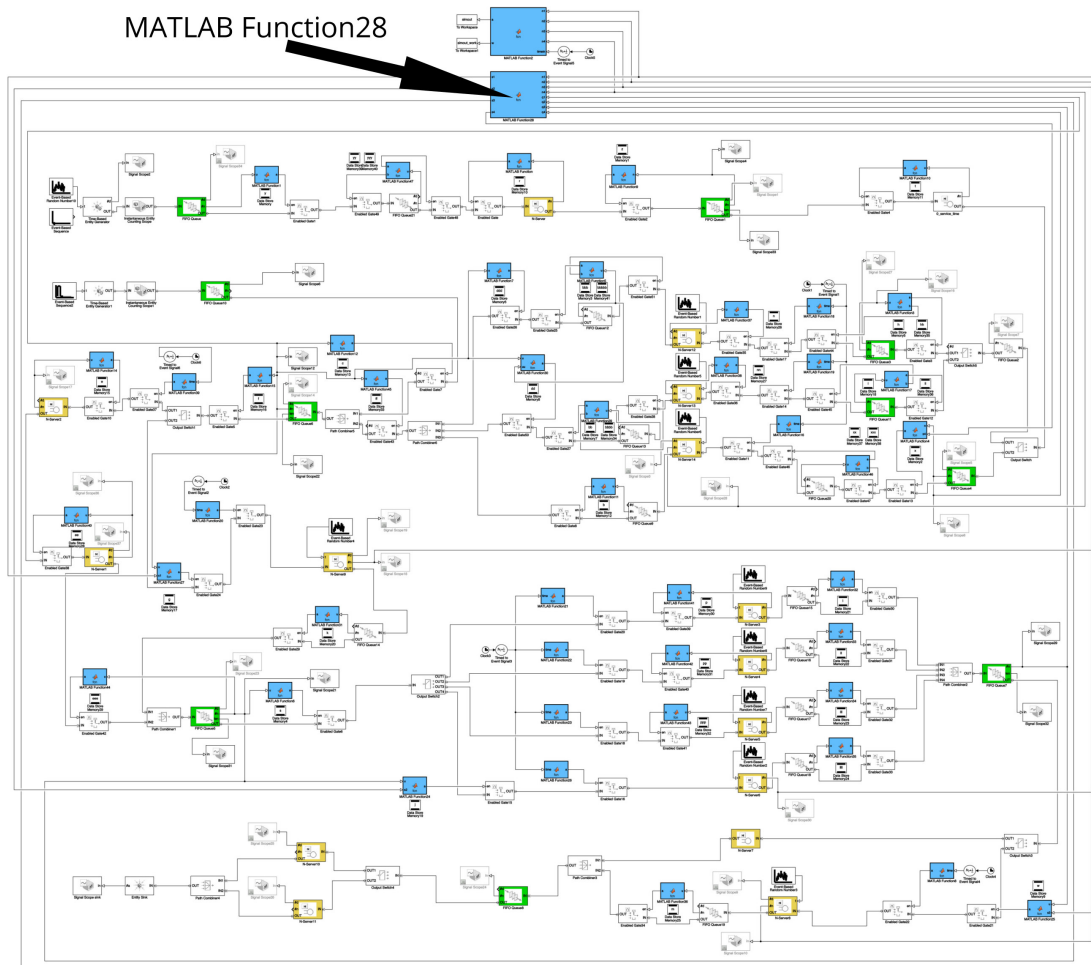
Zmíněná problematika využití jednoho operátora pro čtyři pracovní pozice *S2.3*, *S3.2.1*, *S4.4*, *S5.2* je řešena blokem *MATLAB Function*²⁸, který je na obrázku 4.2 zvýrazněn. Blok funguje na základě vstupních hodnot o délce fronty každému serveru předcházející a informaci, zda je konkrétní server v provozu, a to podle počtu entit v serveru se nacházejících. Na základě výrokové logiky funkce odesílá binární hodnoty do svých subfunkcí. Ty pak řídí samotnou regulaci toku entit do serverů. V případě překročení specifického počtu entit nutného pro spuštění serveru v alespoň jedné frontě, dostane *MATLAB Function*²⁸ požadavek pro odblokování toku entit do serveru. Pomocí prioritního

sestavení funkce systém upřednostňuje požadavky pro odblokování prioritně výše postaveného serveru. Priorita je klesající s tímto pořadím: *S3.2.1*, *S4.4*, *S5.2*, *S2.3*. V případě, že kterákoli vstupní hodnota charakterizující počet entit v serveru je nenulová, funkce blokuje vstupu entit do zbývajících tří serverů, a to i pokud je naplněna fronta serveru s vyšší prioritou. Každý stroj tak pracuje bez přerušení, dokud neklesne počet entit v jemu předcházející frontě na požadovanou hodnotu. Díky tomuto řízení je v jeden časový okamžik vždy v provozu nejvýše jeden server ze čtyř výše zmíněných.

Algoritmus MATLAB Function28

```
function [s321, s44, s52, s23]=fcn(n321, n44, n52, n23,
q4, q5, q6, q33)
if q4>=80 && n44==0 && n52==0 && n23==0
    s321=1, s44=0, s52=0, s23=0
elseif q5>=60 && q4<80 && n321==0 && n52==0 && n23==0
    s321=0, s44=1, s52=0, s23=0
elseif q6>=40 && q5<60 && n321==0 && n44==0 && n23==0
    s321=0, s44=0, s52=1, s23=0
elseif q33>=25 && q6<40 && q5<60 && q4<80 && n321==0 &&
    n44==0 && n52==0
    s321=0, s44=0, s52=0, s23=1
else
    s321=0, s44=0, s52=0, s23=0
end
end
```

Blok To Workspace v zapojení zajišťuje přenos dat ze SimeEvents do Matlabu pro jejich další zpracování. Zaznamenává se čas probíhající simulace a aktivita serverů *S2.3*, *S3.2.1*, *S4.4*, *S5.2*.



Obrázek 4.2: Model výrobní linky

5 VÝSLEDKY SIMULACÍ

Vytvořený model dle stanovených parametrů kapacity front, procesních časů serverů, počtu zpracovávaných entit v jednom stroji najednou, času vyřazení stanoviště z provozu, způsobu příchodu entit do systému apod. simuluje chování reálné linky. Parametry lze libovolně nastavovat a sledovat jejich vliv na výsledek simulace.

Pro demonstraci je dále rozebráno osm simulací s rozdílným nastavením parametrů zájmu. Tabulka 5.1 uvádí základní rozdělení jednotlivých případů. Celkový čas T každé jedné simulace je nastaven na $T = 172800$ s. Délka simulace tak zachycuje chování výrobní linky v časovém úseku 48 hodin. Před spuštěním simulačního procesu se v systému nenachází žádná entita na jakékoli pozici. První entita je vygenerována až v momentě spuštění, tedy $t = t_0 = 0$ s. V samotné simulaci toto rozjetí linky, kde fronty jsou prázdné a servery přijímají entity okamžitě, může vést ke zkreslení výsledků. Proto při vyhodnocování pracovní vytíženosti operátora serverů *S2.3*, *S3.2.1*, *S4.4*, *S5.2* je uvedeno srovnání výsledků dat z časového úseku $T_c \in \langle t_0; T \rangle$ s $T_s \in \langle 86400; T \rangle$ s, kdy je zanedbán časový úsek $t \in \langle 0; 86400 \rangle$ s, což je doba dvaceti čtyř hodin. Výstupy simulací jsou hodnoty vygenerovaných entit primárním a sekundárním zdrojem a množství odchodících entit ze systému, což představuje počet zpracovaných výrobků. Dále průměrný počet entit čekajících ve frontách *Q2*, *Q4*, *Q5*, *Q6*, což jsou sklady předcházející strojům s variabilní pracovní dobou. Počet entit v těchto frontách v průběhu času je znázorněn graficky.

Tabulka 5.1: Rozdělení simulací

Simulace	Case	Procesní časy	Pomocný operátor	Poznámka
1	1	Deterministické	ANO	
2	1	Deterministické	NE	
3	2	Stochastické	ANO	$\sigma = 5\% z \mu$
4	2	Stochastické	NE	$\sigma = 5\% z \mu$
5	3	Stochastické	ANO	$\sigma = 10\% z \mu$
6	3	Stochastické	NE	$\sigma = 10\% z \mu$
7	4	Deterministické	ANO	Časové omezení serveru
8	4	Deterministické	NE	Časové omezení serveru

5.1 Case1

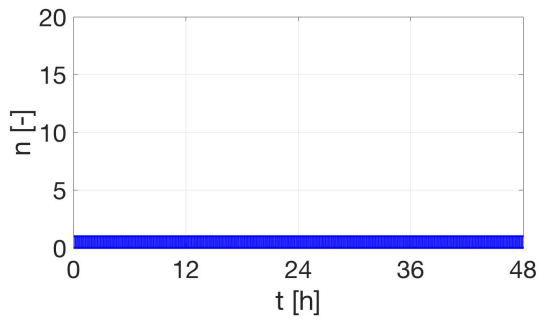
V prvním případě jsou srovnány dvě simulace s deterministicky určenými výrobními časy všech serverů. Simulace 1 popisuje stav, kdy bylo možné zapojit do práce operátora pomocných strojů *S2.3*, *S3.2.1*, *S4.4*, *S5.2*. Naopak simulace 2 sleduje vliv neaktivity těchto pozic po celou dobu simulace. Mimo to měly simulace 1 a 2 stejně nastavené vstupní parametry, které jsou zapsány v tabulce 5.2.

Tabulka 5.2: Vstupní hodnoty pro Case 1

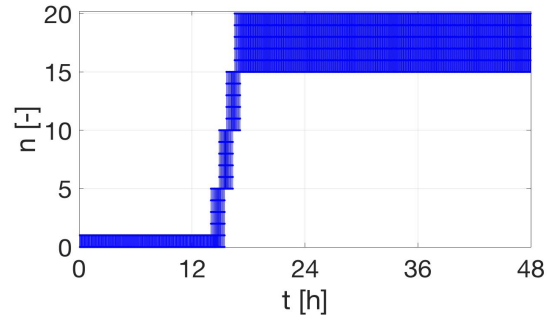
Generátor entit	Časová distribuce	Množství entit	Doba generování [s]
Primární zdroj	Deterministická	1	37
Sekundární zdroj	Deterministická	10	380
Fronta	Druh řazení	Kapacita	
<i>Q1</i>	FIFO	35	
<i>Q2</i>	FIFO	20	
<i>Q3.1, Q3.2, Q3.3</i>	FIFO	30	
<i>Q4</i>	FIFO	120	
<i>Q5</i>	FIFO	200	
<i>Q6</i>	FIFO	100	
<i>Q7</i>	FIFO	80	
Server	Časová distribuce	Počet paralel. serverů	Procesní čas [s]
<i>S1</i>	Deterministická	5	180
<i>S2.1, S2.2, S2.3</i>	Deterministická	5	395
<i>S3.1.1, S3.1.2</i>	Deterministická	20	760
<i>S3.2.1</i>	Deterministická	5	380
<i>S4.1, S4.2, S4.3, S4.4</i>	Deterministická	5	605
<i>S5.1</i>	Deterministická	5	190
<i>S5.2</i>	Deterministická	5	330
<i>S6.1, S6.2</i>	Deterministická	5	360

Vizualizace počtu entit v závislosti na čase ve frontách *Q2*, *Q4*, *Q5*, *Q6* simulace 1 je na obrázcích 5.1, 5.3, 5.5 a 5.7. Pro simulaci 2 jsou tato data vykreslena na obrázcích 5.2, 5.4, 5.6, 5.8. Obrázky vykreslují průběh v celém časovém úseku T_c .

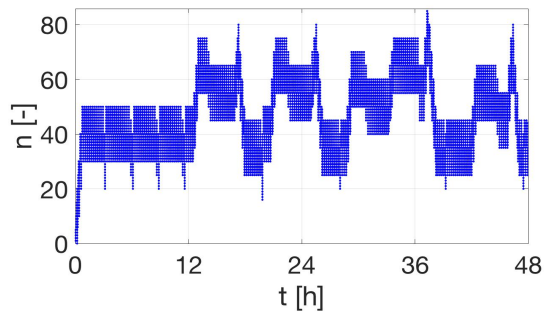
Srovnáním těchto grafů je možno určit vliv nezapojení se pomocného operátora do výrobního procesu. Především u grafů front *Q4*, *Q5* a *Q6* simulace 1 lze pozorovat cyklické chování systému, kdy operátor při kritickém naplnění skladu pomůže se zpracováním entit. Zvýšením propustnosti na jedné pozici tak způsobí zvýšení počtu entit na pozici následující. Pokud i zde množství překročí dovolenou mez, přejde operátor k serveru navazujícímu. Takto se cyklicky u jednotlivých serverů střídá. Naopak z obrázků front *Q2* a *Q5* lze určit, že dojde k zahlcení systému. Servery pozice *S2* a *S4* nestačí entity zpracovávat, čímž dochází k prodlevám.



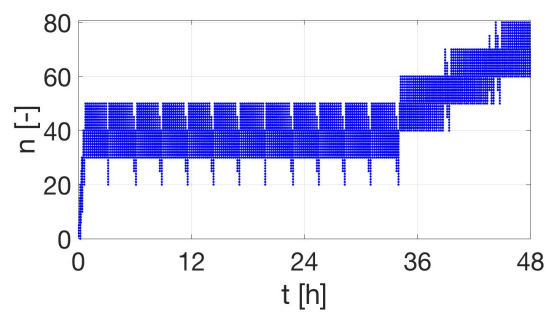
Obrázek 5.1: Simulace 1, Q2



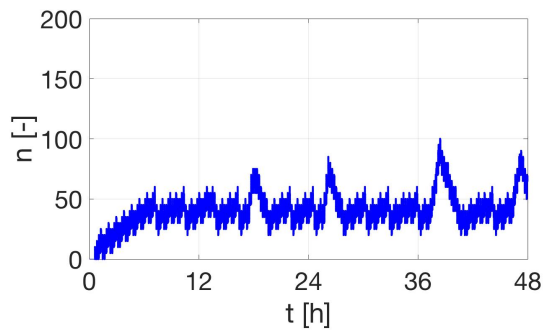
Obrázek 5.2: Simulace 2, Q2



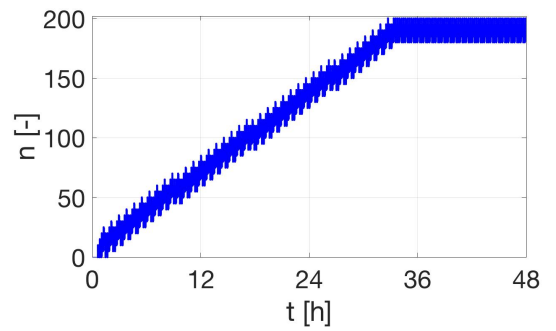
Obrázek 5.3: Simulace 1, Q4



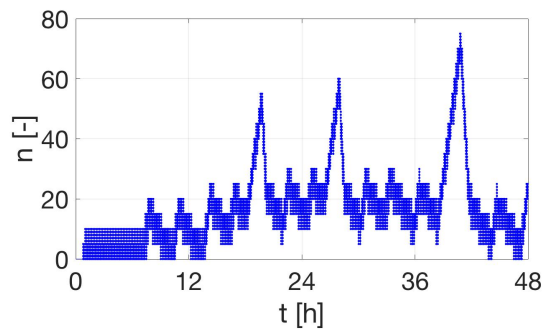
Obrázek 5.4: Simulace 2, Q4



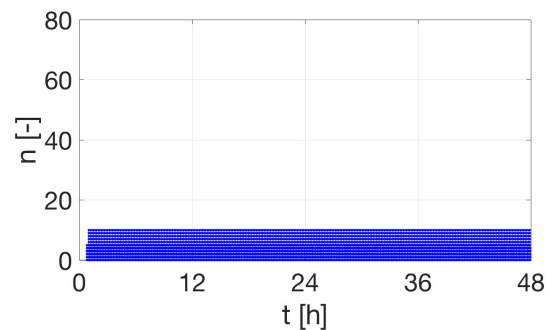
Obrázek 5.5: Simulace 1, Q5



Obrázek 5.6: Simulace 2, Q5



Obrázek 5.7: Simulace 1, Q6



Obrázek 5.8: Simulace 2, Q6

Tabulka 5.3 shrnuje základní data získaná ze simulací 1 a 2. Nulová hodnota L_{Q2} zde může být matoucí v porovnání s obrázkem 5.1, kde lze pozorovat entity ve frontě $Q2$. Vysvětlením pro tento jev je princip discrete-event simulation. Ačkoli tedy entity do fronty vstupují, při volné následující pozici okamžitě odcházejí a doba jejich pobytu ve frontě je nulová.

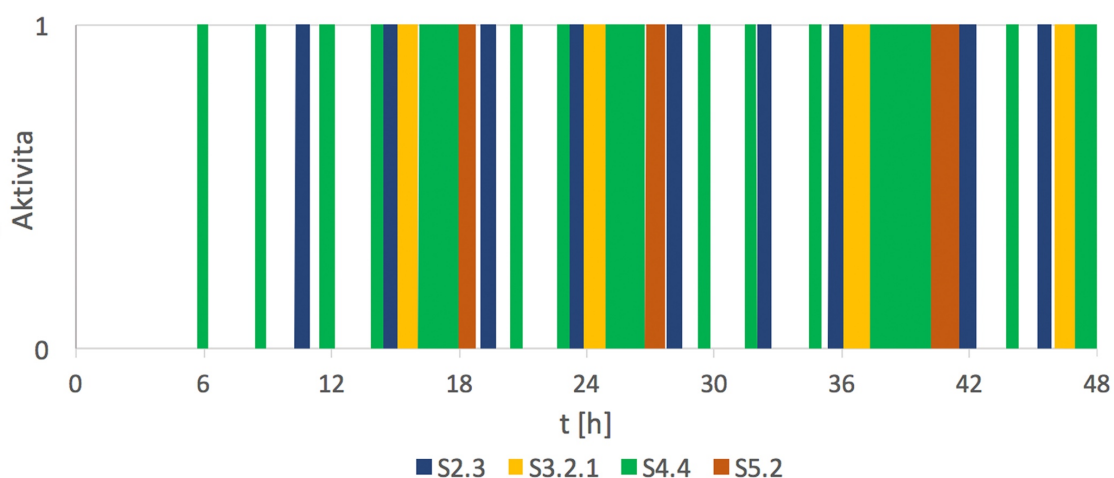
Tabulka 5.3: Výstupní hodnoty pro Case 1

Průměrný počet entit ve frontě	Simulace 1	Simulace 2
L_{Q2}	0,0	12,0
L_{Q4}	47,3	44,5
L_{Q5}	40,4	124,9
L_{Q6}	16,9	3,4
Primární zdroj vygenerováno	4671	4513
Sekundární zdroj vygenerováno	140	175
Odchozí entity	4515	4195

Pracovní vytížení pomocného operátora v simulaci 1 je pro časové úseky T_c a T_s srovnáno v tabulce 5.4. Je uveden celkový čas aktivity a procentuální využití během daného časového úseku. Obrázek 5.9 graficky ilustruje zapojení operátora na jednotlivých pracovních pozicích. Celkový poměr mezi zabarvenou a bílou plochou vystihuje poměr aktivity a neaktivity pracovníka.

Tabulka 5.4: Časové vytížení operátora serverů $S2.3$, $S3.2.1$, $S4.4$, $S5.2$, simulace 1

Časový úsek simulace	Doba aktivity operátora [h]	Využití operátora [%]
T_c	22,3	46,4
T_s	15,0	62,4



Obrázek 5.9: Vytížení operátora, simulace 1

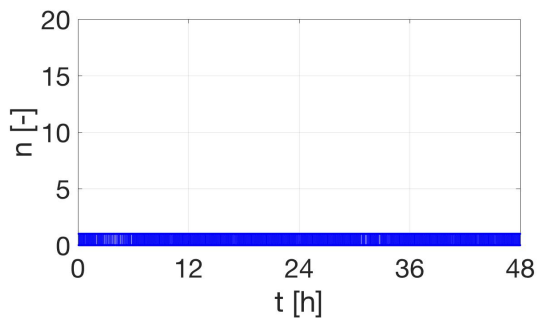
5.2 Case 2

Case 2 srovnává dvě simulace, které mají u strojů s lidskou obsluhou S2.1, S2.2, S2.3, S3.2.1, S4.1, S4.2, S4.3, S4.4 a S5.2 nastaven stochasticky definovaný výrobní čas. U těchto serverů je pro procesní čas jedné entity nastaven s $\sigma = 5\%$ hodnoty μ . Příchozí entity generované primárním zdrojem mají čas příchodu stanoven exponenciální distribucí. Rozdíl mezi simulací 3 a 4 je opět v neaktivitě serverů S2.3, S3.2.1, S4.4, S5.2 po dobu celé simulace.

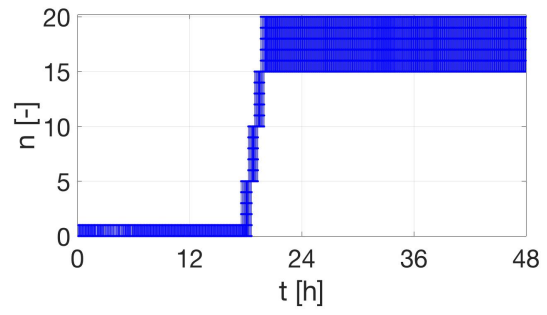
Tabulka 5.5: Vstupní hodnoty pro case 2

Generátor entit	Časová distribuce	Množství entit	Doba generování [s]
Primární zdroj	Exponenciální	1	$\lambda=37$
Sekundární zdroj	Deterministická	10	380
Fronta	Druh řazení	Kapacita	
Q1	FIFO	35	
Q2	FIFO	20	
Q3.1, Q3.2, Q3.3	FIFO	30	
Q4	FIFO	120	
Q5	FIFO	200	
Q6	FIFO	100	
Q7	FIFO	80	
Server	Časová distribuce	Počet paralel. serverů	Procesní čas [s]
S1	Deterministická	5	180
S2.1, S2.2, S2.3	Gaussova	5	$\mu=395, \sigma=19,75$
S3.1.1, S3.1.2	Deterministická	20	760
S3.2.1	Gaussova	5	$\mu=380, \sigma=19,00$
S4.1, S4.2, S4.3, S4.4	Gaussova	5	$\mu=605, \sigma=30,25$
S5.1	Deterministická	5	190
S5.2	Gaussova	5	$\mu=330, \sigma=16,50$
S6.1, S6.2	Deterministická	5	360

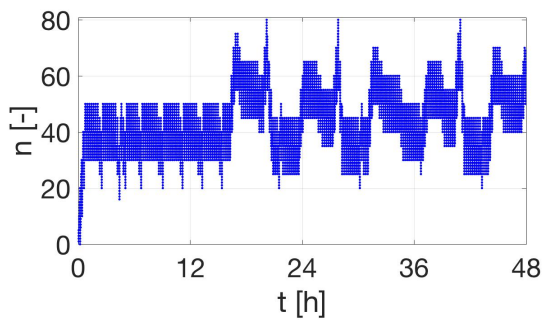
Série obrázků 5.10, 5.12, 5.14, 5.16 znázorňuje počet entit v jednotlivých frontách v průběhu simulace 3. Z těchto ilustrací lze vyčíst, že žádná fronta není nikdy zaplněna. Zároveň u front Q4, Q5 a Q6 je vidět určité cyklické opakování v množství entit v průběhu simulace. V porovnání s nimi jsou obrázky 5.11, 5.13, 5.15, 5.17, které znázorňují vliv neaktivity pomocných serverů v simulaci 4. Stejně jako u simulace 2 lze pozorovat rozdíl u front Q2 a Q5. Ty jsou po určitém časovém úseku zaplněny na svou maximální kapacitu a tím omezují procesy, které jim ve výrobě předcházejí.



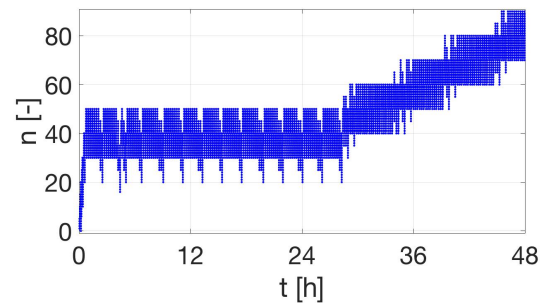
Obrázek 5.10: Simulace 3, Q2



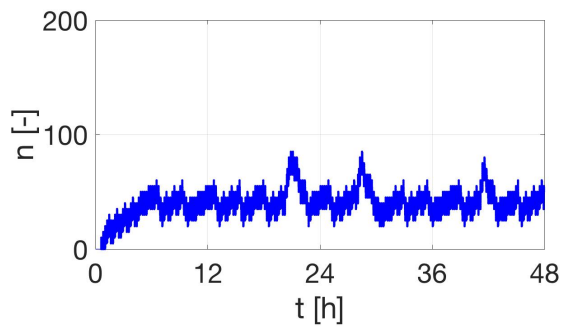
Obrázek 5.11: Simulace 4, Q2



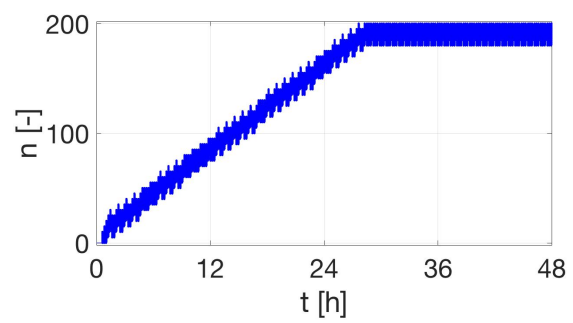
Obrázek 5.12: Simulace 3, Q4



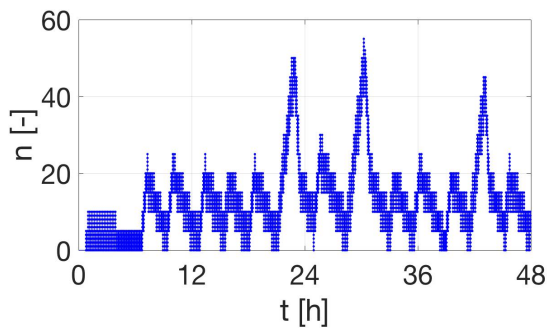
Obrázek 5.13: Simulace 4, Q4



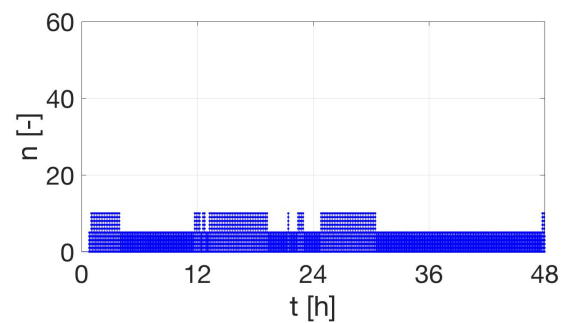
Obrázek 5.14: Simulace 3, Q5



Obrázek 5.15: Simulace 4, Q5



Obrázek 5.16: Simulace 3, Q6



Obrázek 5.17: Simulace 4, Q6

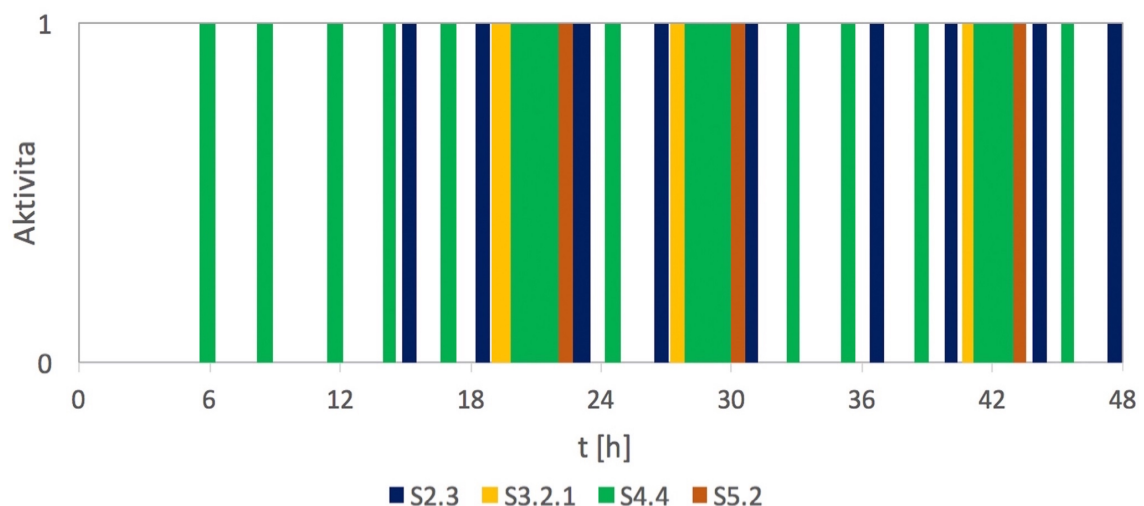
Tabulka 5.6 srovnává výsledky simulací 3 a 4, kde zvýšené hodnoty průměrného počtu entit ve frontě poukazuje na problémy s rychlostí zpracování na navazujících serverech. Hodnota $L_{Q2} = 0$ v simulaci 3 ve srovnání s obrázkem 5.10 má stejné opodstatnění jako v Case 1 pro simulaci 1. Celkovou aktivitu pomocného operátora v simulaci 3 shrnuje tabulka 5.7. Vizualizaci pracovního zapojení na jednotlivých strojích ilustruje obrázek 5.18. Jeho porovnáním s obrázkem 5.9 z Case 1 lze určit, že práce operátora v simulaci 3 je v průběhu času více členitá.

Tabulka 5.6: Výstupní hodnoty pro Case 2

Průměrný počet entit ve frontě	Simulace 3	Simulace 4
L_{Q2}	0,0	11,1
L_{Q4}	43,8	47,8
L_{Q5}	39,7	136,2
L_{Q6}	12,8	2,5
Primární zdroj vygenerováno	4622	4447
Sekundární zdroj vygenerováno	170	200
Odchozí entity	4500	4150

Tabulka 5.7: Časové vytížení operátora serverů S2.3, S3.2.1, S4.4, S5.2, simulace 3

Časový úsek simulace	Doba aktivity operátora [h]	Využití operátora [%]
T_c	20,3	42,4
T_s	12,2	51,0



Obrázek 5.18: Vytížení operátora, simulace 3

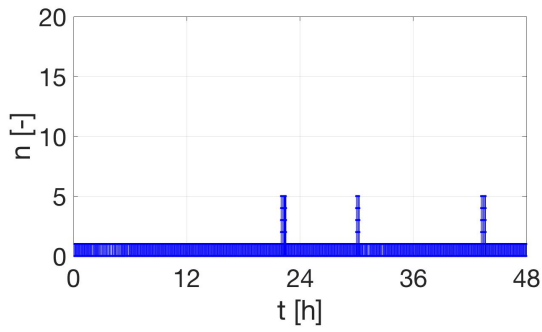
5.3 Case 3

Case 3 srovnává výsledky simulací 5 a 6. Serverům *S2.1, S2.2, S2.3, S3.2.1, S4.1, S4.2, S4.3, S4.4* a *S5.2* jsou přiřazeny stochastické výrobní časy, kde hodnota σ odpovídá deseti procentům hodnoty μ . Primární zdroj generuje entity stochasticky dle exponenciálního rozdělení. V simulaci 6 jsou servery *S2.3, S3.2.1, S4.4, S5.2* neaktivní. Tabulka 5.8 obsahuje zadané parametry simulací 5 a 6.

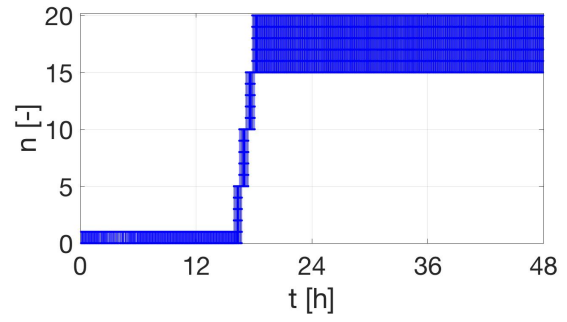
Tabulka 5.8: Vstupní hodnoty pro Case 3

Generátor entit	Časová distribuce	Množství entit	Doba generování [s]
Primární zdroj	Exponenciální	1	$\lambda=37$
Sekundární zdroj	Deterministická	10	380
Fronta	Druh řazení	Kapacita	
<i>Q1</i>	FIFO	35	
<i>Q2</i>	FIFO	20	
<i>Q3.1, Q3.2, Q3.3</i>	FIFO	30	
<i>Q4</i>	FIFO	120	
<i>Q5</i>	FIFO	200	
<i>Q6</i>	FIFO	100	
<i>Q7</i>	FIFO	80	
Server	Časová distribuce	Počet paralel. serverů	Procesní čas [s]
<i>S1</i>	Deterministická	5	180
<i>S2.1, S2.2, S2.3</i>	Gaussova	5	$\mu=395, \sigma=39,5$
<i>S3.1.1, S3.1.2</i>	Deterministická	20	760
<i>S3.2.1</i>	Gaussova	5	$\mu=380, \sigma=38,0$
<i>S4.1, S4.2, S4.3, S4.4</i>	Gaussova	5	$\mu=605, \sigma=60,5$
<i>S5.1</i>	Deterministická	5	190
<i>S5.2</i>	Gaussova	5	$\mu=330, \sigma=33,0$
<i>S6.1, S6.2</i>	Deterministická	5	360

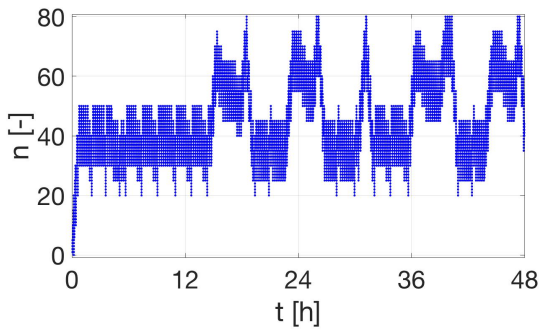
Vizualizaci počtu entit ve frontách v průběhu simulace 5 postihují obrázky 5.19, 5.21, 5.23 a 5.25. Oproti tomu změnu způsobenou trvalou neaktivitou serverů *S2.3, S3.2.1, S4.4, S5.2* v simulaci 6 vykreslují obrázky 5.20, 5.22, 5.24 a 5.26. Na charakteristice počtu entit ve frontě *Q4* ze simulace 6 lze vidět jejich postupný nárůst po 24 hod. od spuštění simulace. Toto je způsobeno plným obsazením fronty *Q5* a nutností serverů pozice *S3* vyčkat na uvolnění prostoru. Z tohoto jevu lze určit, že propustnost za frontou *Q5* je nejnižší, přestože se fronta *Q2* dle obrázku 5.20 zaplní nejrychleji. Důvodem brzkého maximálního naplnění fronty *Q2* je vyšší intenzita generování entit než rychlost zpracování na pozici *S2*.



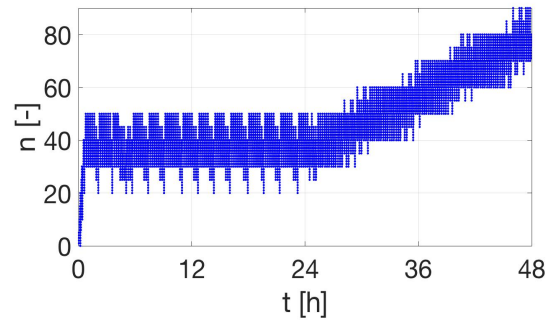
Obrázek 5.19: Simulace 5, Q2



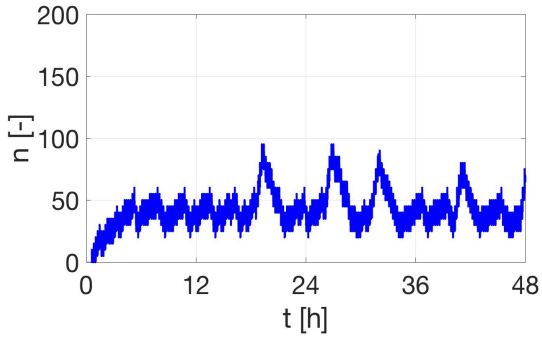
Obrázek 5.20: Simulace 6, Q2



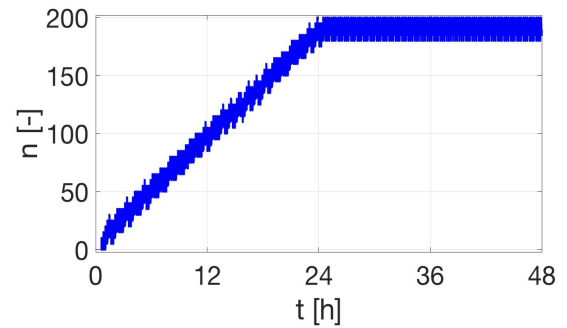
Obrázek 5.21: Simulace 5, Q4



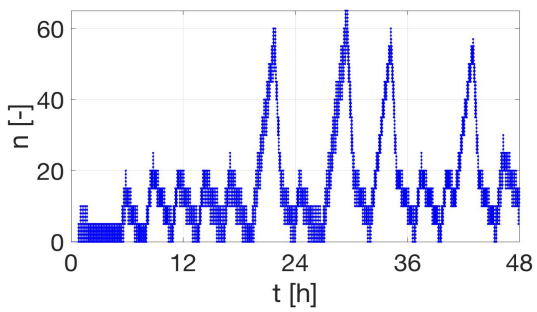
Obrázek 5.22: Simulace 6, Q4



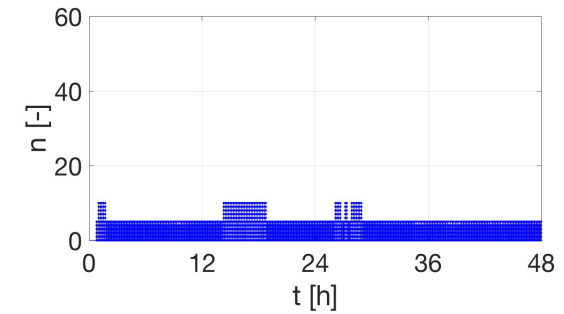
Obrázek 5.23: Simulace 5, Q5



Obrázek 5.24: Simulace 6, Q5



Obrázek 5.25: Simulace 5, Q6



Obrázek 5.26: Simulace 6, Q6

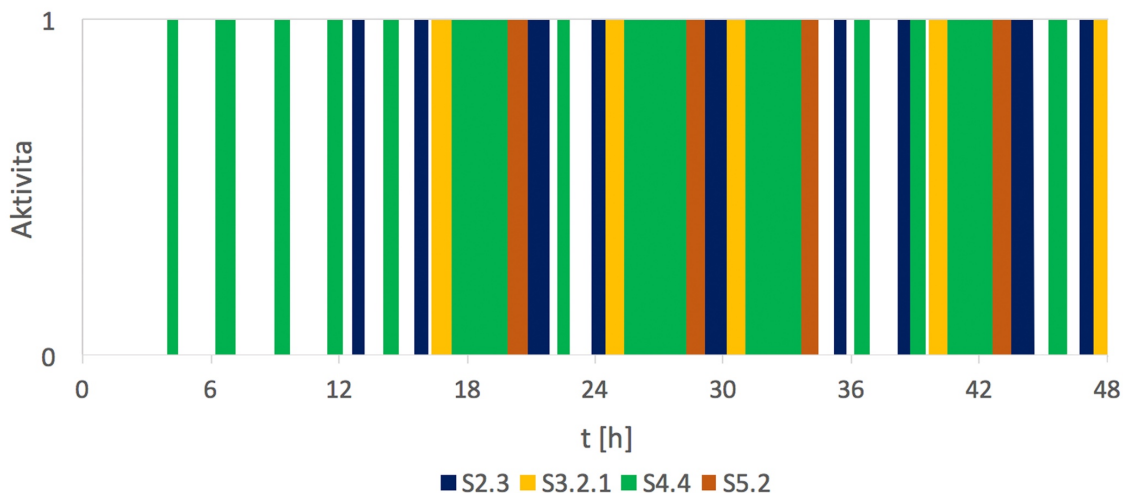
Tabulka 5.9 zaznamenává výstupní data simulací 5 a 6. Informace o zapojení pomocného operátora do výrobního procesu obsahuje tabulka 5.10. Aktivita operátora je graficky znázorněna na obrázku 5.27 a lze vidět, že oproti případům ze simulace 1 a 3 je pracovník mnohem souvisleji zatížen a jeho prostoje mezi nasazením v jednotlivých serverech jsou značně zredukovány.

Tabulka 5.9: Výstupní hodnoty pro Case 3

Průměrný počet entit ve frontě	Simulace 5	Simulace 6
L_{Q2}	0,0	12,1
L_{Q4}	45,2	47,3
L_{Q5}	42,0	144,2
L_{Q6}	15,0	1,7
Primární zdroj vygenerováno	4622	4403
Sekundární zdroj vygenerováno	235	195
Odchozí entity	4540	4100

Tabulka 5.10: Časové vytížení operátora serverů S2.3, S3.2.1, S4.4, S5.2, sim. 5

Časový úsek simulace	Doba aktivity operátora [h]	Využití operátora [%]
T_c	27,8	57,9
T_s	18,3	76,0

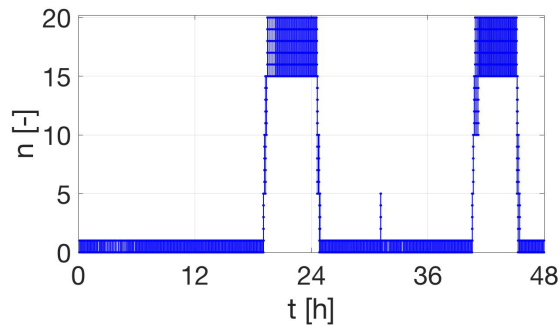


Obrázek 5.27: Vytížení operátora, simulace 5

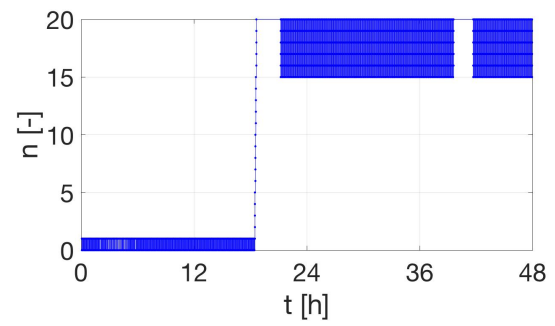
5.4 Case 4

Case 4 zahrnuje simulaci 7 a 8, které modelují dvoufázovou odstávku stroje S3.1.1. Procesní časy všech serverů a generování entit je stanoveno deterministicky. Vstupní parametry odpovídají tabulce 5.2 z Case 1. Server je mimo provoz po dobu tří hodin v náhodném časovém úseku $t \in \langle 62350; 73150 \rangle$ s a následně ještě další dvě hodiny v intervalu

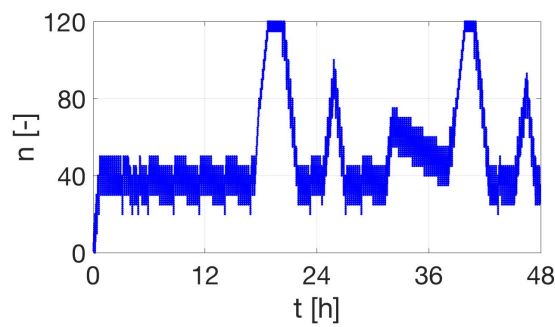
$t \in \langle 140000; 147200 \rangle$ s. Rozdíl mezi simulacemi 7 a 8 je opět v neaktivitě serverů S2.3, S3.2.1, S4.4, S5.2 v simulaci 8.



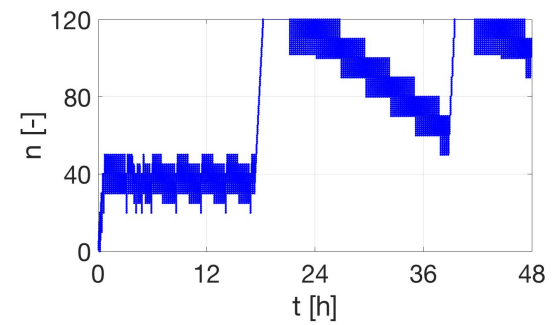
Obrázek 5.28: Simulace 7, Q2



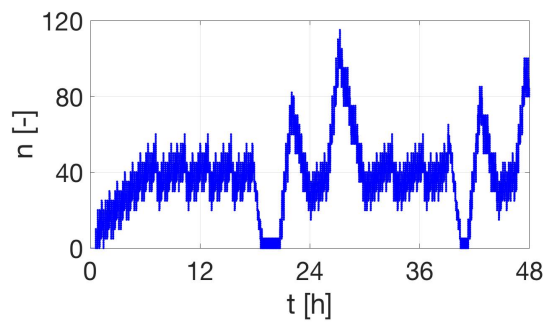
Obrázek 5.29: Simulace 8, Q2



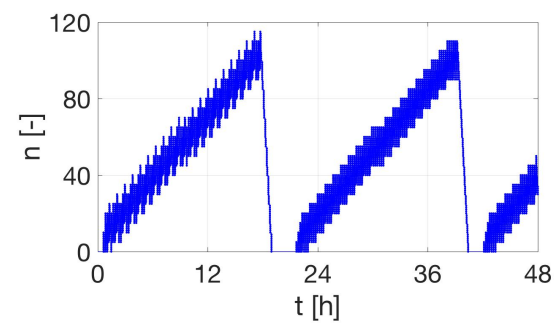
Obrázek 5.30: Simulace 7, Q4



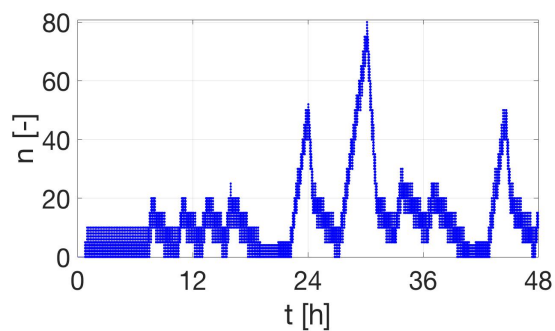
Obrázek 5.31: Simulace 8, Q4



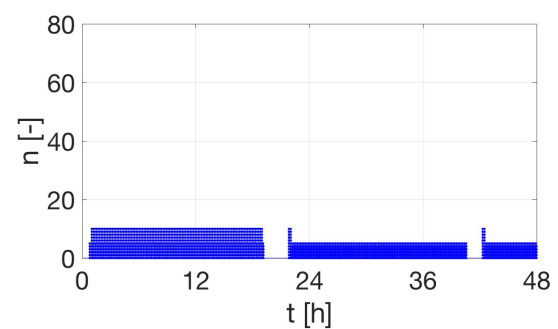
Obrázek 5.32: Simulace 7, Q5



Obrázek 5.33: Simulace 8, Q5



Obrázek 5.34: Simulace 7, Q6



Obrázek 5.35: Simulace 8, Q6

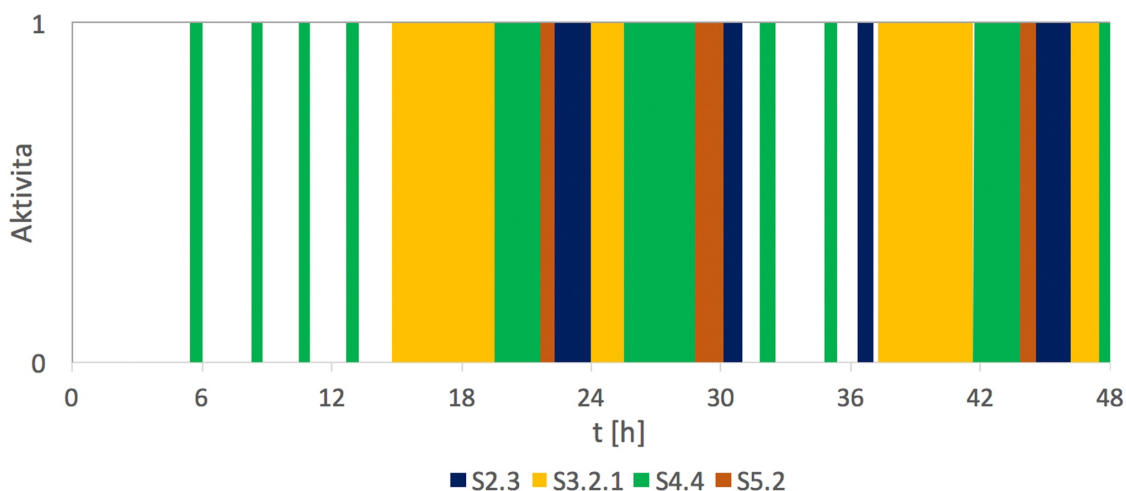
Počet entit ve frontách při simulaci 7 je na obrázcích 5.28, 5.30, 5.32 a 5.34. Ke srovnání jsou uvedeny obrázky 5.29, 5.31, 5.33 a 5.35 simulace 8 ovlivněné neaktivitou pomocného operátora. Na grafické vizualizaci jde vidět vliv odstavení serveru S3.1.1 jako prudkou změnu počtu entit v jednotlivých frontách. Tabulka 5.11 uvádí výstupní hodnoty ze simulací pro Case 4. Hodnoty časového vytížení operátora v průběhu dvou úseků T_c a T_s pro simulace 7 a 8 jsou vypsány v tabulce 5.12. Nasazení pracovníka v jednotlivých serverech je znázorněno na obrázku 5.2. Oproti vizualizacím obrázků 5.9, 5.18 a 5.27 z Case 1, 2 a 3, kde lze pozorovat nejvyšší aktivitu pomocného serveru S4.4, je v případě simulace 7 nejvyšší pracovní zapojení v serveru S3.2.1, který je paralelně zapojen k odstavenému stroji S3.1.1.

Tabulka 5.11: Výstupní hodnoty pro Case 4

Průměrný počet entit ve frontě	Simulace 7	Simulace 8
L_{Q2}	4,0	11,1
L_{Q4}	51,7	77,8
L_{Q5}	38,0	45,4
L_{Q6}	13,1	1,7
Primární zdroj vygenerováno	4538	4044
Sekundární zdroj vygenerováno	175	135
Odchozí entity	4385	3820

Tabulka 5.12: Časové vytížení operátora serverů S2.3, S3.2.1, S4.4, S5.2, sim. 7

Časový úsek simulace	Doba aktivity operátora [h]	Využití operátora [%]
T_c	27,0	56,2
T_s	21,7	78,0



Obrázek 5.36: Vytížení operátora, simulace 7

5.5 Shrnutí

Tabulka 5.13 uvádí nejpodstatnější shrnující informace ze všech osmi simulací. Údaje zde uvedeny, včetně počtu příchozích a odchozích entit, jsou vztaženy pouze pro časový úsek T_s . Sloupec příchozí entity T_s udává celkový počet entit do systému vstupujících v časovém úseku T_s . Jedná se o sumu počtu entit vygenerovaných primárním a sekundárním zdrojem.

Tabulka 5.13: Shrnutí výsledků simulací v časovém úseku T_s

Simulace	Výrobní časy	Pomocný operátor	Využití operátora v časovém úseku T_s [%]	Příchozí entity T_s	Odchozí entity T_s
1	Deretmin.	ANO	62,4	2365	2350
2	Determin.	NE	-	2228	2140
3	Stochastic.	ANO	51,0	2340	2365
4	Stochastic.	NE	-	2195	2120
5	Stochastic.	ANO	76,0	2390	2455
6	Stochastic.	NE	-	2140	2095
7	Determin.	ANO	78,0	2310	2335
8	Determin.	NE	-	2007	1995

Srovnáním hodnot odchozích entit během časového úseku T_s pro stejné vstupní hodnoty dle Case s rozdílným nastavením aktivity pomocného operátora je možno pozorovat nárůst produktivity celé výrobní linky způsobený jeho zapojením tohoto pracovníka. Konkrétně u deterministických vstupních hodnot Case 1 činí rozdíl vyprodukovaných entit za 24 hodin 210 kusů, což představuje 8,9 % denní produkce. Pro stochastické rozdělení dle Case 2 a Case 3 jsou rozdíly 10,1 % a 10,5 % mezi odchozími entitami simulací s pomocným operátorem a bez něj. V případě dočasného odstavení serveru *S3.1.1* na tři a následně na dvě hodiny je produktivita linky bez operátora o 14,7 % nižší než v případě jeho nasazení. Zároveň z porovnání hodnot výstupních entit simulace 1 a simulace 7, které měly kromě odstávky serveru stejně nastavené parametry, lze určit, že se produktivita snížila o 0,6 %. Oproti tomu u simulace 8 lze pozorovat pokles 6,7 % vyprodukovaných entit v porovnání se simulací 2, které opět mají kromě nastavené odstávky všechny vstupní parametry stejné. Na srovnání těchto dvou případů jde vidět, že při vhodně naplánovaném odstavení strojů může být produktivita celé linky téměř nedotčena. Zároveň lze s delší dobou odstávky očekávat výrazně prohlubující se rozdíl mezi oběma simulacemi.

Při porovnání počtu odchozích entit ze systému v souvislosti s deterministickými a stochastickými výrobními časy lze pozorovat u simulace 1 rozdíl 0,6 % a 4,2 % v počtu zpracovaných entit oproti simulacím 3 a 5, tedy při simulacích s přítomným pomocným operátorem. Simulace 2 vykazuje o 0,9 % a 2,1 % odlišné výsledky oproti simulacím 4 a 6. Z těchto výsledků lze usoudit, že pro případ, kdy je míra náhodnosti dostatečně malá a konkrétně při výše nastavených vstupních parametrech je σ do 5 % hodnoty μ , lze simulaci bez zásadní chyby modelovat jako deterministickou.

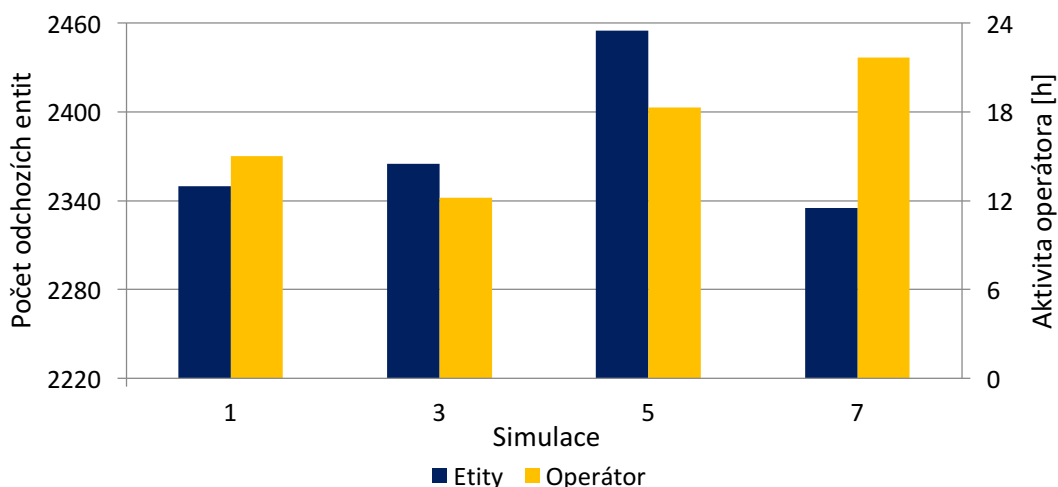
Naopak vytíženost operátora serverů *S2.3, S3.2.1, S4.4, S5.2* se při deterministických a stochastických výrobních časech mění o desítky procent. U simulace 1 je využit z 62,4 % času T_s a jeho zapojení do pracovního procesu je dle Obrázek 5.9 pravidelné a cyklické. Při simulaci 3 je časové využití operátora 51,0 %, ale i přesto celkový počet odchozích entit převyšuje o patnáct simulaci 1. V simulaci 5 je operátor zapojen do výroby 76,0 % doby T_s . Díky jeho vyššímu pracovnímu nasazení lze vysvětlit nárůst vyprodukovaných entit v tomto případě. Obrázek 5.37 graficky znázorňuje počet odchozích entit ze systému v průběhu doby T_s pro simulace 1, 3, 5 a 7. Zároveň je vykreslena časová vytíženost operátora pomocných serverů. V navazujících simulacích tedy pro zjištění zatížení pracovníka není možné zanedbat stochastický vliv bez značně zkreslených výsledků tohoto parametru.

Hodnoty příchozích entit do systému vykazují u Case 1, 2 a 3 maximální rozdíl 25 entit pro simulace s operátorem a 88 entit pro simulace bez operátora v časovém úseku T_s . Avšak více než rozdílem mezi stochastickým a deterministickým příchodem jsou tato data ovlivněna stavem za linkou, protože v případě zablokování výstupu z generátoru se produkce entit pozastaví. Tím je celkový význam modelování stochastického příchodu entit do systému minimální. Nižší hodnoty u simulací Case 4 jsou způsobeny právě zpomaleným chodem výrobní linky.

Srovnáním všech grafických výstupů zaznamenávajících počet entit ve frontě pro Case 1, 2 a 3 lze vypořádat podobný charakter vývoje v průběhu simulace pro případy se zapojením operátora do výroby a bez jeho zapojení. Generálně z toho jde usoudit, že deterministické či stochastické výrobní časy mají na průběh výroby mnohem menší vliv než změna způsobená aktivním operátorem navíc.

Pro vyhodnocení faktu, zda se z ekonomického hlediska pomocného operátora vyplatí zaměstnat lze využít rovnice (5.1). Pokud $x > 1$, je finančně výhodné tohoto pracovníka najmout a s narůstající hodnotou x se zvyšuje i ekonomická výhodnost.

$$x = \frac{\text{Hodnota produktů vyrobených navíc za jednu hodinu}}{\text{Hodinová mzda operátora}} \quad (5.1)$$



Obrázek 5.37: Vybrané výstupní hodnoty

ZÁVĚR

Pro zajištění konkurenceschopnosti na trhu je pro průmyslové podniky téměř nezbytné držet krok s pokrokovým vývojem Průmyslu 4.0. V jednotlivých krocích jsou tak implementovány do výroby prvky splňující charakteristické rysy Industry 4.0. Jedním z nich je i virtualizace, což je schopnost propojení reálného systému s virtuálním modelem. Tím vzniká možnost předvídat a efektivně plánovat výrobní procesy. Jedním ze způsobů, jak vytvořit model virtuální továrny, je s využitím discrete-event simulation založené na pojmu teorie front.

Právě na principu discrete-event simulation pracuje program SimEvents, což je součást programu Matlab-Simulink. Software SimEvents byl využit k vytvoření simulačního modelu výrobní linky. Model je výstupem této práce a jeho sestavení je založeno na topologii reálné výrobní linky firmy BOSCH DIESEL s.r.o. – Jihlava. Pomocí různých vstupních parametrů lze modelem simulovat množství situací, které mohou ovlivňovat celkový výrobní proces. Mezi tyto případy patří možnost simulovat odstávky zvolených výrobních strojů, nastavení deterministických či stochastických procesních časů serverů a nastavení aktivity pomocných pracovišť. Mezi výstupy, které je možné zaznamenávat, patří délka front, průměrná doba čekání, vytíženost serverů a pomocného operátora nebo množství příchozích a odchozích entit. S pomocí těchto dat tak může model sloužit pro pilotní simulace k určení tzv. úzkých míst ve výrobě a ve výsledku až ke změně topologie výrobní linky.

S modelem je možno dále pracovat a rozvíjet jej pro zajištění dalších funkcí. Mezi možnosti rozšíření patří schopnost simulovat zpracovávání více druhů entit výrobní linkou. Jedním ze způsobů, jak toho dosáhnout, může být využití předem nastaveného atributu entity, který určí logiku chodu linkou a procesní časy v jednotlivých serverech. Jako další možnost rozvoje se nabízí rozšíření pro simulování komplexnějších odstávek více serverů najednou nebo možnost zapojení více než jednoho operátora obsluhujícího pomocné servery pro zvýšení produktivity. Mimo rozšíření modelu z hlediska možností nastavení vstupních parametrů je žádoucí doprogramovat i přívětivější uživatelské rozhraní tak, aby i do problematiky nezavázaný uživatel byl schopen model používat. K tomuto účelu Matlab nabízí možnost vytvoření tzv. GUI. Propojení s programem Microsoft Excel, odkud lze zadaná data nahrát do jednotlivých bloků simulace, je rovněž ideální způsob pro zjednodušení a urychlení práce s modelem.

Software SimEvents nabízí značnou variabilitu ve vytváření modelů, a to především díky možnosti doprogramovat vlastní matlabovské kódy, které následně řídí logiku chodu entit při simulaci. Mezi nevýhody tohoto programu patří nemožnost měnit parametry modelu v průběhu simulace. Nelze tedy po spuštění chodu simulace zasahovat a libovolně měnit nastavení podle aktuálního vývoje. Zároveň je pro použití modelu v komerční sféře nutné zakoupení licence na software Matlab, Simulink a SimEvents. Částka za všechny tyto programy nemusí být pro menší podniky zanedbatelná.

Použitím simulací reálných systémů lze zajistit zvýšení produktivity továrny, snížení nákladů výroby, zlepšení bezpečnosti a pracovních podmínek operátorů. Pomocí získaných dat lze omezit plýtvání přírodními zdroji a zredukovat celkovou energetickou náročnost výroby. To může vést ke snížení produkce skleníkových plynů a jiných dopadů na životní prostředí.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] MAREK, P. *Implementace chytré továrny*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 80 s. Vedoucí práce doc. Ing. Vladislav Škorpil, CSc.
- [2] CHAOUCHI, H. *The internet of things: connecting objects to the web*. London: Hoboken: ISTE; Wiley, 2010, 265 s. ISBN 978-1-84821-140-7.
- [3] GILCHRIST, A. *Industry 4.0: the industrial internet of things*. New York: Apress, 2016, 250 s. ISBN 978-1-4842-2046-7.
- [4] MAŘÍK, V. *Průmysl 4.0: výzva pro Českou republiku*. Praha: Management Press, 2016, 262 s. ISBN 978-80-7261-440-0.
- [5] VITÁK, J. *Simulátor výrobních linek*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 53 s. Vedoucí diplomové práce Ing. Jan Pásek, CSc.
- [6] JACOB, M. Discrete event simulation. *Resonance*. India: Springer-Verlag, 2013, 18(1), s. 78-86. DOI: 10.1007/s12045-013-0010-x. ISSN 0971-8044.
- [7] PRATEEK, S. Discrete-Event Simulation. *International Journal of Scientific & Technology Research*. IJSTR, 2015, 4(4), s. 136-140. ISSN 2277-8616.
- [8] FISHMAN, S. G. *Discrete event simulation: modeling, programming and analysis*. New York: Springer, 2001, 537 s. ISBN 0-387-95160-1.
- [9] ŠEDA, M. *Modely hromadné obsluhy* [online]. Článek, Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011, [cit. 2018-05-12]. Dostupné z: http://web2.vslg.cz/fotogalerie/acta_logistica/2011/2_cislo/3_seda.pdf
- [10] GROSS, D. *Fundamentals of queueing theory*. 4th ed. Hoboken: John Wiley, 2008, 500 s. ISBN 978-0-471-79127-0.
- [11] HALAS, D. *Optimalizace výrobních procesů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 71 s. Vedoucí diplomové práce RNDr. Pavel Popela, Ph.D.
- [12] NELSON, R. *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. New York: Springer, 2010, 583 s. ISBN 978-1-4419-2846-7.
- [13] KARPÍŠEK, Z. *Matematika IV: statistika a pravděpodobnost*. 4., přeprac. vyd. Brno: Akademické nakladatelství CERM, 2014, 171 s. ISBN 978-80-214-4858-2.
- [14] JANÍKOVÁ, D. *Model pro ekonomickou simulaci procesů (sledování nákladů na nízkou jakost)*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 107 s. Vedoucí dizertační práce doc. Ing. Alois Fiala, CSc.
- [15] HORKÝ, M. *Modely hromadné obsluhy*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 59 s. Vedoucí diplomové práce prof. RNDr. Ing. Miloš Šeda, Ph.D.
- [16] KLEINROCK, L. *Queueing systems. Volume I, Theory*. New York: John Wiley, 1975, 417 s. ISBN 0-471-49110-1.

- [17] GRAY, M. A. Discrete Event Simulation: A Review of SimEvents. *Computing in Science & Engineering*. USA: IEEE, 2007, 9(6), s. 62-66. DOI: 10.1109/MCSE.2007.112. ISSN 1521-9615.
- [18] LI, W., R. MANI a P. J. MOSTERMAN. Extensible Discrete-Event Simulation framework in SimEvents. In: *Winter Simulation Conference (WSC), 2016*. IEEE, 2016, s. 943-954. DOI: 10.1109/WSC.2016.7822155.
- [19] THE MATHWORKS, INC. *SimEvents Getting Started Guide*. R2015b, 90 s.
- [20] Pricing and Licensing. *MathWorks* [online]. Natick, USA, 2018, [cit. 2018-05-23]. Dostupné z: <https://www.mathworks.com/pricing-licensing.html>

SEZNAM PŘILOH

K práci je přiloženo CD obsahující:

- elektronickou verzi bakalářské práce ve formátu PDF
- SimEvents: simulace_1.slx
- SimEvents: simulace_2.slx
- SimEvents: simulace_3.slx
- SimEvents: simulace_4.slx
- SimEvents: simulace_5.slx
- SimEvents: simulace_6.slx
- SimEvents: simulace_7.slx
- SimEvents: simulace_8.slx