Simulator model methodology, version 2.5

Revision 1.5

Denis Albert, Peter Gagarinov PhD,
Maxim Kuvshinov PhD, Ilya Roublev PhD,
Vadim Zakroischikov

Allied Testing LTD

16th of Oct, 2015

1 Introduction

This document is devoted to the methodology of construction and testing of the Simulator model.

1.1 What is Exchange Simulator?

The main purpose of Exchange Simulator is to model reaction of market participants on user intrusions (activity that is absent in historical data files). Simulator replays historical orders flow, processes orders submitted by users according to exchange rules and models the market response on these user orders. The latter means it may vary characteristics of historical orders and (if necessary) may generate additional order activity. Simulator consists of state of the art probability-driven empirically justified models generating realistic behavior of the market and performs the following actions:

- replays historical orders flow;
- processes user intrusions according to exchange rules;
- models reaction of market participants on user intrusions, i.e. influence on historical orders flow, as well as generates orders counter flow not present in historical data files.

1.2 Range of use

Exchange Simulator could be useful for solving the following problems:

- Back testing of intra-day trading strategies;
- Trading against historical scenarios;
- Construction of sophisticated, instrument-specific quantitative models of instrument behavior;
- Continuous, event-driven replication and simulation of order books;
- Testing of market connectivity;
- Assistance for configuration of new exchanges.

1.3 Scope of simulation

The simulation could be run only for one stock for periods within the continuous trading session.

The supported types of orders for simulation (intrusion) are only plain limit and market orders (iceberg orders are not supported). The supported types of events are submissions, cancellations, updates. However we do not generate simulated updates (updates that are not present in historical data file) because an update event could be presented as successive submission and cancellation. Thus, we generate only simulated submissions and cancellations.

1.4 Terminology

Below we introduce the terms and definitions we use in the article:

- *Historical Data (HD)* — basic historical market data. Currently Level III data are needed.
- *User Orders (UO)* — orders generated by user trading strategies (reaction of market which we are interested in).
- *Historical orders (HO)* — orders that are taken from historical data files. These historical orders are replayed by Simulator. It is possible that some attributes of these orders would be changed during simulation (for example, Elasticity Model could change price limits of historical limit orders according to changes of market state due to user intrusions), thus, original events for historical orders are modified and generate one category of simulated events.
- *Simulated Orders (SO)* — orders that are generated by Simulator models according to the expected reaction of market participants on user intrusions (these orders are purely model generated and not present in historical data files).
- *Distance between price levels* — the difference between prices of limit orders placed on these levels.
- *Aggressive orders* — orders generating trades at the instant of their submission (for example, all market orders are aggressive).
- *Order book* — pair of ordered (possibly, empty) finite sets that depend on time and describe for ask and bid sides all price levels for these sides and all volumes of limit orders with limit prices on these levels (formal definition and all notions concerning order book see in section 2).
- *Order book snapshot* — pair of ordered (possibly, empty) finite sets that depend on time and describe for ask and bid side all price levels for these sides and total aggregated volumes of limit orders with limit prices on these levels (formal definition and all notions concerning order book snapshot see in section 2).
- *Buckets* — sequence of non intersecting intervals for grouping the data, i.e. for the sample $\{x_i\}_{i=1}^N$ all buckets form the set of intervals $\{I_k\}_{k=1}^{N_b}$ such that $x_i \in \cup_{k=1}^{N_b} I_k$ for all $i = 1, \dots, N$, where for numbers $x_b^1 \leqslant x_e^1 \leqslant x_b^2 \leqslant \dots \leqslant x_e^{N_b}$ we let $I_1 = [x_b^1, x_e^1]$ and for all $k = 2, \dots, N_b$ we denote $I_k = [x_b^k, x_e^k]$ if $x_b^k > x_e^{k-1}$ and $I_k = (x_b^k, x_e^k]$ otherwise.

2 Basic definitions and acronyms

- t — real-time in seconds (currently Simulator works on per second time grid, so all instants are rounded up to seconds, see section 5.3 for details). It is assumed that $t \in [t_{min}, t_{max}]$, where $[t_{min}, t_{max}]$ is the time interval describing continuous trading session (auctions are excluded, see section 5.3 for details). If it is not pointed explicitly we assume for each factor Φ depending on time that $\Phi(t)$ denotes the value of this factor at the end of second t. Below for all the notions that depend on t we assume that if these notions are used in the text when t is omitted, then t is the current time instant.

- ϖ_{min} and ϖ_{max} — minimal and maximal values of price, $0 < \varpi_{min} < \varpi_{max}$.
- ϖ — the set of all possible values of price for plain limit and market orders (see section 5.3 for details), ϖ is an ordered set numbered by integers

  $$\varpi = \{\varpi_i \mid i \in \mathbb{Z}\}, \qquad \varpi_{i_1} < \varpi_{i_2}, \quad i_1, i_2 \in \mathbb{Z}, i_1 < i_2.$$

  We should note that in reality the price takes values from $[\varpi_{min}, \varpi_{max}]$, but we allow for prices to be out of this interval, even to take negative values, just to simplify notations. In fact, all orders taken from historical data files have prices from $[\varpi_{min}, \varpi_{max}]$, moreover, we do not allow for user orders to have prices out of the mentioned interval. Thus, when $p \in \varpi$ and $p \notin [\varpi_{min}, \varpi_{max}]$, then formally volumes on the price level p is zero both in historical and simulated order books (see below definitions of $V(OB(t), price)$, $OB^H(t)$ and $OB^S(t)$), thus, difference between these order books on the price level p would be also zero. Hence, the constraint that price must be from $[\varpi_{min}, \varpi_{max}]$ would be fulfilled automatically (see section 7.5 for details).

  Moreover, in current implementation it is assumed, that $\varpi_{i+1} - \varpi_i$ does not depend on $i \in \mathbb{Z}$, i.e. we have constant tick size (see section 5.3 for details).
- $\iota(p)$ — number of price level for given price p, i.e. $\iota(p) = i \in \mathbb{Z}$ such that $p = \varpi_i$.
- $\delta\varpi$ — tick size, $\delta\varpi = \varpi_{i+1} - \varpi_i$.
- $\{\Delta\}$ — the set of all possible values of distance between different prices from ϖ, i.e. in our case of constant tick size $\{\Delta\} = \{j \cdot \delta\varpi \mid j \in \mathbb{Z}\}$.
- $e = (id, t, V, p, side, type, origin) \in E$ — market event (here E denotes the set of all possible events) that are characterized by 7 attributes:
  - id — unique identifier of order, positive integer value;
  - t — time instant when event occurred;
  - V — volume, positive integer value;
  - $p \in \varpi \cap [\varpi_{min}, \varpi_{max}]$ — price;
  - $side \in \{ask, bid\}$;
  - $type \in \{subm, canc, trd\}$ — event type, i.e. whether event is submission ($subm$), cancellation ($canc$) or trade (trd) [1];

[1] We should note that that we do not consider below so-called *update* events (that simply modify the attributes of the corresponding order), because we present each of them as successive cancellation and submission (for the latter new id is assigned). Besides, some historical orders may be on the market during several days. We emulate this by means of submission events that occur at start of each day *before* all events within both continuous trading session and open auction. Thus, some preprocessing of historical events should be done to consider the model described in this document.

- $\overline{OBS}^{H}_{bid}(t, t_{width})$ — the definition is analogous to the previous one with replacement 'ask' on 'bid'.

- $\overline{OBS}^{S}_{ask}(t, t_{width}) = \overline{OB}^{S,dist}_{ask}(t, t_{width})$, i.e. $\overline{OB}^{dist}_{ask}(t, t_{width})$ is applied to simulated order book $OB(t) = OB^S(t)$.

- $\overline{OBS}^{S}_{bid}(t, t_{width})$ — the definition is analogous to the previous one with replacement 'ask' on 'bid'.

- $\rho(OB^H(t), OB^S(t))$ — measure between historical and simulated order books (snapshots) at the moment t (here we use definition of $\rho(OB^1(t_1), OB^2(t_2))$ given above).

- $\rho^{+,ask}(OB^H(t), OB^S(t))$ — measure between ask side of historical and ask side of simulated order books at the moment t taking into account only positive difference on price levels (here we use definition of $\rho^{+,ask}(OB^1(t_1), OB^2(t_2))$ given above).

- $\rho^{+,bid}(OB^H(t), OB^S(t))$ - the definition is analogous to previous one with replacement of 'ask' on 'bid'.

- $\rho^{-,ask}(OB^H(t), OB^S(t))$ - measure between ask side of historical and ask side of simulated order books at the moment t taking into account only negative difference on price levels (here we use definition of $\rho^{-,ask}(OB^1(t_1), OB^2(t_2))$ given above).

- $\rho^{-,bid}(OB^H(t), OB^S(t))$ — the definition is analogous to previous one with replacement of 'ask' on 'bid'.

2.1 Main Flows

Main flows of Exchange Simulator are:

- Submission flows (bid/ask) of nonaggressive orders (historical/simulated/user);

- Cancellation flows (bid/ask) of orders (historical/simulated/user)

- Trade flows (bid/ask) (submissions of aggressive orders historical/simulated/user);

We do not consider update flows because update events are presented as successive cancellations and submissions.

3 The concept of model

Let us consider model of Exchange Simulator.

3.1 Model blocks and interconnection between them

There are seven model blocks (below they themselves would be called models, but in any case they are only parts of the general model of Exchange Simulator taken as a whole):

1. Duration/Intensity Model

2. Submission Price Level Model

3. Elasticity/Aggressiveness Model

4. Diming Model

5. Cancellation Model

6. Trade Model

7. Order Volume Model

Let us describe the algorithm of how Simulator does function.

Duration Model (1) generates the next moment of event. It could be submission, cancellation or trade event (there are no simulated update events during simulation since we present update events as successive cancellation and submission events).

For submission event we consider the nature of event, it could be historical (generated according to historical submission), simulated (generated according to model) or user order (generated according to user intrusion). For user order we just reproduce user intrusion, because all characteristics of order are known. For historical order we change historical price limit of limit order according to response of market due to user intrusions; to generate price limit for historical limit order we use our Elasticity/Aggressiveness Model (3) — it models elasticity of price limits of entry orders on change of market state. For simulated order there are three reasons of submission:

1. Ordinary limit order activity as reaction on market intrusion.

   Here we choose limit price according to Submission Price Level Model (2) and volume of order—to Order Volume Model (7).

2. Diming case (simulated orders submission activity ahead extremely large limit order near best price levels)

   Here we generate characteristics of orders by means of Diming Model (4).

3. Counter flow of aggressive orders (simulated market order activity).

   For aggressive simulated orders we choose their characteristics from Trade Model (6). This model is responsible for market orders and generates their volumes.

For cancellation event we should select order to cancel (any order except user ones could be chosen). To do this we use Cancellation Model (5).

The composite model is implemented as

- batch process for estimation of the model block parameters for a given stock by using a sample of full order history data file (we recommended to use at least two weeks of input data for highly liquid stock and one month data for illiquid stocks);

- runtime process responsible for simulation. This process uses the parameters that are estimated on historical data by means of batch process described in the preceding item.

4 The problem

Suppose we have the system representing ideal market, which operates only in **continuous** trading phase (i.e. without auctions) with

- plain limit orders,
- market orders

and supports only submissions, cancellations and trades, assuming that updates (modifications of some attributes of given orders) are presented as subsequent cancellations and submissions.

As inputs to our system, we have:

- the sequence $\mathcal{E}^{in} = \{e_k^{in}\}_{k=1}^{N_{in}}$ of input historical events (see footnote 1 on page 7 for details), $e_k^{in} \in E$ — historical event occurred at time $t_k^{in} = t(e_k^{in})$, $origin(e_k^{in}) = H$, $k = 1, \ldots, N_{in}$, t_k^{in} is non-decreasing in k;
- user intrusion in the form of some **trading strategy** (its exact definition would be given below) that basically generates a sequence of events $\{e_l^{usr}\}_{l=1}^{N_{usr}}$, $e_l^{usr} \in E$ — user-generated event at time $t_l^{usr} = t(e_l^{usr})$, $type(e_l^{usr}) \neq trd$, $origin(e_l^{usr}) = U$, $l = 1, \ldots, N_{usr}$, t_l^{usr} is non-decreasing in l.

Our task is to replicate adequate reaction of market to user strategy through generation of additional flow of events that being united with modified historical events and user-generated events form output flow $\mathcal{E}^{out} = \{e_m^{out}\}_{m=1}^{N_{out}}$, e_k^{out} — output event at time $t_m^{out} = t(e_m^{out})$, $origin(e_m^{out}) \in \{H, S, U\}$, $m = 1, \ldots, N_{out}$, t_m^{out} is non-decreasing in m. Here we mean that $\{e_m^{out}\}$ includes all (possibly, modified) events from $\{e_k^{in}\}$ and all events from $\{e_l^{usr}\}$.

5 Solution approach to the problem

5.1 Algorithm working during real-time

Within this section we would use mainly not per second time scale that was introduced above, but the scale connected to indices of events. The reason for this is that several events may be occurred within single second, but the algorithm described below should process these events successively.

Thus, we assume, that all processed historical events and all currently simulated events have indices expressed by natural numbers. Introduce the following notations, that would be used below.

- $k_{min} = \min\{k = 1, \ldots, N_{in} : t_k^{in} \geqslant t_{min}\}$ — index of the first historical event within continous trading session (here t_{min} denotes the start time of continuous trading session, see also section 2).
- κ, $\kappa \in \{k_{min} - 1, \ldots, N_{in}\}$, — the index of last processed historical event.
- μ, $\mu \in \{k_{min} - 1, \ldots\}$, — the index of last generated simulated event.

$\mathcal{E}^i = \{e^i_k\}_{k=1}^{N^i}$ of historical events from the corresponding historical data file. Let $\breve{N}^0 = 0$, $\breve{N}^i = \sum_{j=1}^i N^j$ for $i = 1, \dots, N$, and $\breve{N} = \breve{N}^N$, $\breve{\mathcal{E}} = \{e_k\}_{k=1}^{\breve{N}}$, where $e_k = e^i_{k-\breve{N}^{i-1}}$ for $k = \breve{N}^{i-1} + 1, \dots, \breve{N}^i$, $i = 1, \dots, N$.

Then Simulator obtains π^i_{elast} and π^i_{dim} and by

$$\pi^i_{elast} = f^b_{elast}(\mathcal{E}^i), \tag{5.15}$$

$$\pi^i_{dim} = f^b_{dim}(\mathcal{E}^i). \tag{5.16}$$

$$\tag{5.17}$$

If the stock under consideration is high liquid (i.e. sufficient number of events per day is available) then

$$\pi^i_{orvol} = f^b_{orvol}(\mathcal{E}^i), \tag{5.18}$$

$$\pi^i_{trd} = f^b_{trd}(\mathcal{E}^i). \tag{5.19}$$

If the stock is low liquid, then for all $i = 1, \dots, N$

$$\pi^i_{orvol} = f^b_{orvol}(\breve{\mathcal{E}}), \tag{5.20}$$

$$\pi^i_{trd} = f^b_{trd}(\breve{\mathcal{E}}). \tag{5.21}$$

Usually, high and low liquid stocks are differed by comparison of $\breve{N}$ with some fixed threshold $\breve{N}_{treshold}$. If $\breve{N} \leqslant \breve{N}_{treshold}$, then the stock is assumed to be low liquid, otherwise it is high liquid.

In any case for all $i = 1, \dots, N$

$$\pi^i_{dur} = f^b_{dur}(\breve{\mathcal{E}}), \tag{5.22}$$

$$\pi^i_{spl} = f^b_{spl}(\breve{\mathcal{E}}), \tag{5.23}$$

$$\pi^i_{canc} = f^b_{canc}(\breve{\mathcal{E}}). \tag{5.24}$$

After these parameters for each model are estimated, it is possible to use them in real-time, i.e. to apply the algorithm from section 5.1 for each day d_i, $i = 1, \dots, N$, assigning $\pi_{mdl} = \pi^i_{mdl}$ for all $mdl \in \{elast, dur, spl, dim, orvol, trd, canc\}$. Or it is possible to use parameters estimated for some day d_i within the mentioned algorithm performed on another day, possibly, not from D, only if the latter day is in some neighborhood of D.

5.3 Assumptions of model

1. Each stock is considered separately and independently from other stocks.

2. All time instants are on per second time grid (all original time instants are rounded up to seconds). Values of all factors at each time instant are taken at the end of the corresponding second.

3. Real-time and batch processes are considered only within continuous trading session, auctions are excluded.

4. Any intrusion is dissolved over time. (!reference to Almgren, impact!) In essence, all user intrusions should not differ significantly from average characteristics of submissions taken for historical orders, large shocks are to be excluded.

5. The price of submissions is changed according to elasticity coefficient which expresses portion from difference between simulated mid-price and historical mid-price. The change of submission price also depends on distance between historical mid-price and historical submission price (see section 7.4 on elasticity model).

6. For each simulated event probability distribution of its time depends on event type and the side of the order book on which it is to occur and does not depend on price and volume of the corresponding order. Probability distribution of order price depends on event type, side and time and does not depend on order volume. Probability distribution of order volume does depend on event type, side, time and price. Thus, these basic characteristics of simulated events may be generated successively and separately.

7. All possible values of prices and price limits at any time instant form discrete set that does not depend on time. Tick size, that is difference between adjacent prices, does not depend on these prices, i.e. tick size is constant.

8. The duration between events (after removing seasonality), cumulative volume, bid-ask spread, volatility of cumulative volume, volatility of return exhibit stable behavior (see section 7.3 on duration model).

9. Probability of submission on some price level does not depend on time and depends only on the values of such factors as textbfdistance from the opposite best price for coming limit order and **difference** between aggregated and average aggregated (averaging is performed in time within some moving window) volumes at that distance. Also, the effect produced by these factors **are statistically independent** (see section 7.5 on submission model).

10. Probability of cancellation of some order does not depend directly on time and depends only on the values of such factors as **relative distance** of this order, **initial distance** of its placement, its **life time** (i.e. period between submission and current time), **difference** between aggregated and average aggregated (averaging is performed in time within some moving window) volumes on the price level corresponding to limit price of this order. Also, the effects produced by these factors **are statistically independent** (see section 7.6 on cancellation model).

7.2.2 Description of Model

We calculate empirical cumulative distribution function of entry volume for historical aggressive orders for both sides $F_a^{ask}(\cdot)$ and $F_a^{bid}(\cdot)$ (exact description of how these distributions are estimated, see in subsection 7.2.4 below).

If it is necessary to choose volume of simulated market order on the side given by *side*, we generate this volume according to random quantity with distribution determined by cdf $F_V(\cdot)$ obtained from $F_a^{side}(\cdot)$ due to the following constraint. If the volume that has been generated is greater than the volume $\bar{V}$ needed for compensation of the difference between historical and simulated order books, the precise volume $\bar{V}$ needed for compensation is generated instead of the initial one.

7.2.3 Inputs and outputs

We have as inputs *side* that determines the side on which market order is to be generated, exact volume $\bar{V} \in [0, +\infty)$ necessary for compensation of the difference between historical and simulated order books (see (5.14)) as well as vector of model parameters π_{trd}, single output is $F_V(\cdot)$, that is cumulative distribution function of volume for historical aggressive orders. The resulting trade volume is a random quantity with distribution given by F_V.

7.2.4 Batch. Estimation of Model

Let $\mathcal{E}^{in} = \{e_k^{in}\}_{k=1}^{N_{in}}$ be the sequence of events that is input argument of the function f_{trd}^b given in (5.19) and (5.21). Let us describe how f_{trd}^b is implemented. Fix some $side \in \{ask, bid\}$. We take

$$\begin{aligned} K_a^{side} = \{k = 1, \dots, N_{in} \mid \\ id(e_k^{in}) \in I_{aggr}, t(e_k^{in}) \in [t_{min}, t_{max}], side(e_k^{in}) \neq side, type(e_k^{in}) = trd\}. \end{aligned}$$

where I_{aggr} is determined by (7.1). Let F_V^{side} be empirical cdf of volumes taken from $\{V(e_k^{in}) \mid k \in K_a^{side}\}$. Then cdf F_a^{side} is calculated via kernel smoothing density estimate for cdf F_V^{side} taken for given number of points N_V (this is done by `ksdensity` function from MatLAB, see [32]). To be more precise, we perform smoothing for probability density function corresponding to $v = \ln(V)$ instead of V itself due to the fact that V may be sufficiently good approximated by lognormal distribution ([31]), hence, the distribution of $\ln(V)$ is more symmetric. After this we determine cdf for $\exp(v) = V$.

The functions $F_a^{ask}(\cdot)$ and $F_a^{bid}(\cdot)$ obtained by this way form the vector of model parameters $\pi_{trd} = \{F_a^{ask}(\cdot), F_a^{bid}(\cdot)\}$ that is output argument of the function f_{trd}^b (see (5.19) and (5.21) above).

7.2.5 Real-time

Let *side* and $\bar{V}$ be fixed and

$$F_V(x) = \begin{cases} 0, & x \leqslant 0, \\ F_a^{side}, & 0 < x < \bar{V}, \\ 1, & x \geqslant \bar{V}. \end{cases}$$

In result, the function f_{trd}^{rt} from (5.13) is defined as

$$f_{trd}^{rt}(side, \bar{V}, \pi_{trd}) = F_V(\cdot).$$

7.3 Duration model

7.3.1 Purpose of model

The duration model serves for generation of time instants for submission of plain limit and market orders and for cancellation of limit orders.

7.3.2 Description of model. Factors for duration

In this section we would like to describe in general the model, all exact definitions and methods to calculate different parameters are in subsections 7.3.5 and 7.3.7.

Thus, we have six flows of events (aggressive orders are excluded):

- submissions on ask side;
- submissions on bid side;
- cancellations on ask side;
- cancellations on bid side;
- trades on ask side;
- trades on bid side.

Every flow is considered separately and independently from other flows. It should be noted that the side for each trade is determined unambiguously, because each trade is generated by one aggressive and one nonaggressive order placed on different sides, but aggressive orders are excluded, see above.

We have the following set of descriptive exogenous variables:

- $qVol^{ask}(t)$ and $qVol^{bid}(t)$ — total cumulative volumes of unmatched orders (i.e. those that are in order book) on ask and bid sides, respectively;
- $sp(t)$ — bid-ask spread;
- $\sigma(qVol^{ask})(t)$ and $\sigma(qVol^{bid})(t)$ — volatilities of total cumulative volumes;
- $\sigma(t)$ - volatility of mid price return.

Let us fix some $side \in \{ask, bid\}$ and introduce the following column vectors of exogenous variables

$$\mathcal{F}^{side}(t) = (\ln(qVol^{side}(t)), \ln(sp(t)), \ln(\sigma(qVol^{side})(t)), \ln(\sigma(t)))^T.$$

Here and below T denotes transposition. We should note that the methods of calculation of $\mathcal{F}^{side}(t)$ in batch and in real-time essentially differ and described in the corresponding subsections (7.3.5 and 7.3.7).

Let us also fix some $type \in \{subm, canc, trd\}$ and determine the value of the index $j = 1, \dots, 6$, corresponding to the current flow of events determined by $side$ and $type$ exactly by the same way as coordinates of the vector $t_{ref}(\mu)$ (see section 5.1). Consider ACD model for this flow, namely, let $\{t_l^j\}$ be the sequence of time instants when events of given type of given side occurred and denote by $\mathcal{F}^j_l$ one of $\mathcal{F}^{ask}(t^j_l)$ and $\mathcal{F}^{bid}(t^j_l)$ depending on what side was fixed. Denote by

$$\Delta t^j_l = t^j_l - t^j_{l-1}$$

the duration between arrival of $(l-1)$-th and l-th events.

Before introducing ACD model we remove seasonality from duration data (so called *diurnal patterns*) by the following way:

$$\widetilde{\Delta t}^j_l = \frac{\Delta t^j_l}{s^j(t^j_l)}. \tag{7.3}$$

Here $s^j(t)$ is some function which simulates deterministic diurnal pattern (see (7.10)). Currently $s^j(t)$ is estimated on the basis of the Nadaraya-Watson smoothing method (see section 7.3.5 for details).

Then log-ACD model for durations (see [17, 16]) is as follows:

$$\begin{aligned} \widetilde{\Delta t}^j_l &= \frac{1}{M^j}\cdot\exp(\psi^j_l)\cdot\varepsilon^j_l, \\ \psi^j_l &= w^j + a^j\cdot\psi^j_{l-1} + b^j\cdot\ln(\widetilde{\Delta t}^j_{l-1}) + (\delta^j)^T\cdot\mathcal{F}^j_{l-1}. \end{aligned} \tag{7.4}$$

Here $\psi^j_l = \mathbb{E}(\widetilde{\Delta t}^j_l|\widetilde{\Delta t}^j_{l-1},\ldots,\widetilde{\Delta t}^j_1;\mathcal{F}^j_{l-1})$, ε^j_l are i.i.d. (their distribution is discussed below), $M^j = \mathbb{E}(\varepsilon^j_l)$, $w^j, a^j, b^j \in \mathbb{R}$, $\delta^j \in \mathbb{R}^4$ (that are parameters of log-ACD model) as well as parameters of distribution of ε^j_l given below are estimated by batch process. The condition of stationarity and invertibility of the process given by (7.4) is $|a^j + b^j| < 1$.

It is assumed that all members of the sequence $\{\varepsilon^j_l\}$ have some predefined distribution (usually Weibull, Gamma or Burr distributions are used). In statistical literature intensity (that is inverse with respect to duration) is often expressed in terms of hazard function. A lot of articles say that hazard functions of Weibull and Gamma distributions are quite flexible for modeling duration. The hazard function of Weibull distribution is monotonous, for Gamma distribution it can take various patterns including so called U-shape pattern.

The probability density function (pdf) for Weibull distribution is

$$f_{\varepsilon^j}(c|\gamma^j) = (\gamma^j)^{\gamma^j-1}\exp{(-c)^{\gamma^j}}, \tag{7.5}$$

where $\gamma^j > 0$. The mean value of Weibull distribution is equal to $M^j = \mathbb{E}(\varepsilon^j) = \Gamma(1 + 1/\gamma^j)$.

The pdf for Gamma distribution is

$$f_{\varepsilon^j}(c|\alpha^j, k^j) = \frac{\alpha^j c^{k^j\alpha^j-1}}{(\lambda^j)^{k^j\alpha^j}\Gamma(k^j)}\exp\left[-\left(\frac{c}{\lambda^j}\right)^{\alpha^j}\right]. \tag{7.6}$$

Here $\lambda^j = \Gamma(k^j)/\Gamma(k^j + 1/\alpha^j)$, $k^j > 0$, $\alpha^j > 0$. The mean value of Gamma distribution is equal to $M^j = \mathbb{E}(\varepsilon^j) = 1$.

Suppose that we estimated all the parameters of duration model for given *type* and *side*. Then simulation of durations is performed as follows. Let

$$g^j(\widetilde{\Delta t},\psi,\mathcal{F}) = w^j + a^j\cdot\psi + b^j\cdot\ln(\widetilde{\Delta t}) + (\delta^j)^T\cdot\mathcal{F}. \tag{7.7}$$