VOLUME 2 NUMBER 6 ○ 1992 ISSN 1210-0552

International Journal on Neural and Mass-Parallel Computing and Information Systems

Editor: M. Novák

VSP IDG

REAL–TIME TRADING MODELS FOR FOREIGN EXCHANGE RATES

O.V. Pictet, M.M. Dacorogna, U.A. Müller, R.B. Olsen, J.R. Ward [1]

Abstract: A set of real-time trading models offering analysis of foreign exchange (FX) rate movements and providing explicit trading recommendations is presented. These models are based on the continuous collection and treatment of FX quotes from market makers operating around the clock. These data are processed by a distributed system of computers performing the tasks of data collection, data validation, indicator computation, rule-based analysis, communication and display generation. The out-of-sample performance of these models is typically close to 18% return per year with unleveraged positions and excluding any interest gains. Diversifying the exposure through a portfolio of currencies reduces the risk of using such models for real trading. With a portfolio of three equally weighted FX rates the maximum drawdown is reduced from an average of 9% to 5% for a single trading model while keeping the annual return practically unchanged.

Key words: *Traiding models, exchange rates*

Received: September 30, 1992
Accepted: October 20, 1992

1. Introduction

A prudent speculator believes he knows how prices will evolve. However, many market-makers also believe that markets are inherently efficient and hence that prices cannot be forecast. Although these are interesting questions, the purpose of this paper is not to discuss market efficiency. Rather, we wish to show that with a reasoned approach and high-quality data, it is possible to design practical and profitable trading models. Indeed, we have developed our own trading models which we present here. These anticipate price movements in the foreign exchange (FX) market sufficiently well to be profitable in the long term yet with acceptable risk behavior.

[1] O.V. Pictet, M.M. Dacorogna, U.A. Müller, R.B. Olsen, J.R. Ward
Olsen & Associates, Research Institute for Applied Economics, Seefeldstrasse 233, 8008 Zürich, Switzerland

A clear distinction should be made between a price change forecast and an actual trading recommendation. A trading recommendation naturally includes some kind of price change forecast, but must also account for the specific risk profile of the dealer or user of the respective trading model. Another distinction is that a trading model must take into account its past trading history while a price forecast is not biased by a position the trading model might be in. A trading model thus goes beyond predicting a price change: it must decide if a certain action has to be taken. This decision is subject to the specific risk profile, the trading history and institutional constraints such as business hours. This paper shows how these different parameters can be integrated into the decision-making process and how they are important for constructing practical models for professional traders.

Our trading models offer real-time analysis of FX-rate movements and generate explicit trading recommendations. These models are based on the continuous collection and treatment of FX quotes by market makers around the clock (up to 5000 non-equally spaced prices per day for the German mark agains the US dollar).

Our models follow the FX market and imitate it as closely as possible. They do not deal directly but instead instruct human FX dealers to make specific trades. In order to imitate real-world trading accurately, they take into account transaction costs in their return computation, generally avoid trading outside market working hours and avoid trading too rapidly. In short, these models act realistically in a manner that a human dealer can easily follow.

In the next section, the FX market is described together with the data that we collect and the actual implementation of the models in our real-time information system. In section 3 we present how a model, given a set of indicators, makes its decisions and what the rules governing these decisions are. The system architecture and the specially developed computer language forming the basis of the trading model implementation are also explained here. Section 4 describes the indicators used by the model to determine in which direction prices are heading. In section 5 we discuss the optimization procedure and introduce a new performance measure that includes a risk component. We analyze the model performance in section 6 by considering various risk measures such as the maximum drawdown or the profit over loss ratio and discuss the distinction between in and out-of-sample data. In the same section we also compare the results obtained with our model to a simple 20 days moving average model and show the importance of taking into account not only the total return but also the risk behavior of the model. Moreover, we show that the risk associated with these models can be considerably reduced by splitting the capital among a portfolio of currencies. Our conclusions are presented in section 7.

a and Processing Environment

change rate is the price at which one national currency can be ex-
or another. The most common currency value notion is the bilateral rate (or simply the FX rate) quoted by a FX trader or reported by endor. This is a nominal exchange rate because it is the number of currency offered in exchange for one unit of another. For example, an marks for 1 US dollar. The spot exchange rate is a particular ex-
a nominal bilateral exchange rate where the transaction takes place ely. Another example is the forward exchange rate where the price now but the transaction takes place in the future. In this paper we only deals with spot exchange rates.

w describe how the FX market operates, what are the data we collect we process them.

FX market and the data sources

price is composed of two quantities: a price for the bid offer (termed rice) and a price for the ask offer (the ask price). The bid and ask major financial institutions are conveyed to dealers' screens by quote such as Reuters, Telerate or Knight Ridder. Deals are negotiated one. The FX market operates globally and around the clock. Any aker may submit new bid/ask prices at any time and many larger ns have branches worldwide so that they can partake in continuous ng.

ting the real world requires a system that collects all available price nd that reacts in real-time to price movements. For our trading e have used mainly Reuters data but other data suppliers provide formation in their FX quotes. Using software developed in-house, t, validate and store price quotes in our database for future use. Each tick in the database contains the following items:

time t_j in GMT[3] (Greenwich Mean Time) at which this price has n recorded in our database,

bid price p_j^{bid},

ask price p_j^{ask}.

name of the bank which issued the price,

location of the particular bank,

l the result of our validation filter.

ndex j identifies individual database records. Reuters pages provide rice p_j^{bid} as a complete number, usually with five digits; p_j^{ask} is given with only the last two digits and is recomputed in full by our software. The granularity of the time steps t_j is 2 seconds. The validation filter information is computed by the filtering algorithm described in [1].

The tick frequency has varied since the beginning of our data collection. Currently the tick frequency for the USD/DEM is approximately 5000 ticks per business day, approximately 3000 ticks per day for the other major rates (USD/JPY, USD/CHF and GBP/USD) and around 1500 ticks per day for the minor rates (USD/FRF, USD/NLG and USD/ITL). Altogether, our database presently contains more than 12 million ticks for each of these rates.

2.2. Trading hours and market holidays

Although the FX market operates continuously, individual traders or institutions generally partake of this market only for a portion of each day. Our models accommodate such users by incorporating the notion of business hours and holidays.

Every trading model is associated with a local market that is identified with a corresponding geographical region. In turn, this is associated with generally accepted office hours and public vacation days. The local market is defined to be open at any time during office hours provided it is neither a weekend nor a public holiday. Our trading models presently support the Zürich, London and Frankfurt markets and it is straightforward to extend this set. Typical opening hours for a model are between 8:00 and 17:30 local time, the exact times depending on the particular local market.

Except for closing an open position if the price hits a stop loss limit (described in section 3.4), a model may not deal [2] outside opening hours or during holidays.

2.3. System overview

Our system, the Olsen Information System (OIS), performs all actions required to operate a set of real-time trading models. These include data collection, data validation, generation of trading recommendations, communication of these recommendations to user-agents residing at our customers and graphical presentation of these on the user-agents. Fig. 1 provides an example of how a trading model looks on a customer's user-agent.

The most significant item displayed on a user-agent is the model's present position or trading recommendation. The O&A trading models currently make gearing recommendations of S−1.0, S−0.5 (short positions), 0 (neutral),

[2] In this paper, we speak of a trading model "dealing" or "entering a new position" but, as noted previously, our trading models do not deal directly but instead recommend a human dealer to make trades. In our pursuit of realism, however, we consider the decision-making process of our models to be just as valid as any made by a "real-life" dealer. In this respect we are deliberately loose in our phraseology.

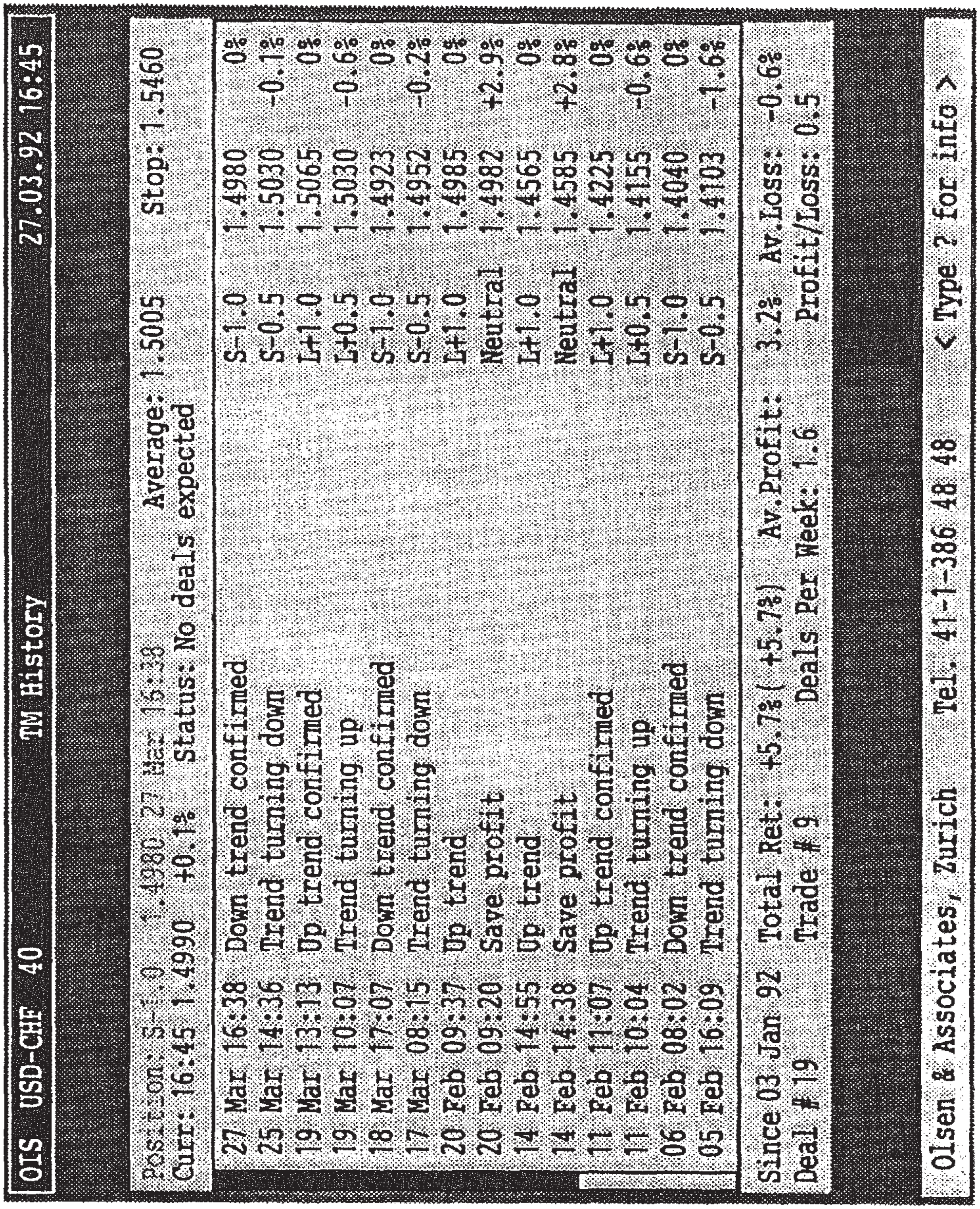

Fig. 1 *A trading model history page. The top section shows the position in which the trading model presently finds itself. The current (most recent) price is also shown here as is a percentage figure indicating the performance since the last deal (current return). The middle section shows a list of previous deals which may be scrolled up or down. The bottom section gives statistics collected since the trading model was started.*

L+0.5 and L+1.0 (long positions). For example, a gearing of S−0.5 means "go short with half the available capital".

2.4. Distributed architecture

The OIS is not constructed as a single huge program with all the required functionality residing in that one entity. Rather it is constructed as a collection of separate programs. In computer-speak, the system is said to be a distributed system because the various programs run in parallel on several computers.

Each such program performs one logical operation. For instance, a program termed the GTrader takes in collected and validated price data and uses this data to produce trading model recommendations. (Several trading models may reside independently of one another within one GTrader invocation). Other programs in the system include a price collector for receiving data from the quote-vendors and a price database manager for storing this data for future use. Splitting up the system into several logical components confers several benefits :

- A failure in one program is localized and does not affect other programs in the system. This increases the overall reliability. For example, if a GTrader program should crash — hopefully, a rare occurrence but one which may happen as a result of hardware failure — collection of raw price data remains unimpaired.
- If a computer should fail, the programs running on that computer may be transferred to another machine. Again, this promotes reliability.
- The performance of the system may be enhanced by adding extra hardware and moving some existing programs on to the new hardware.
- Having separate programs perform logically distinct operations helps simplify the design and structure of the system. The modularity of the implementation is vouchsafed.

3. The Decision Process of the O&A Trading Models

3.1. Overview of the trading model implementation

The overall structure and data-flow of an O&A trading model is depicted in Fig. 2.

Indicator computations form a central part of an O&A trading model providing an analysis of past price movements. (A detailed analysis of the indicators is given in section 4). These indicators are mapped into actual

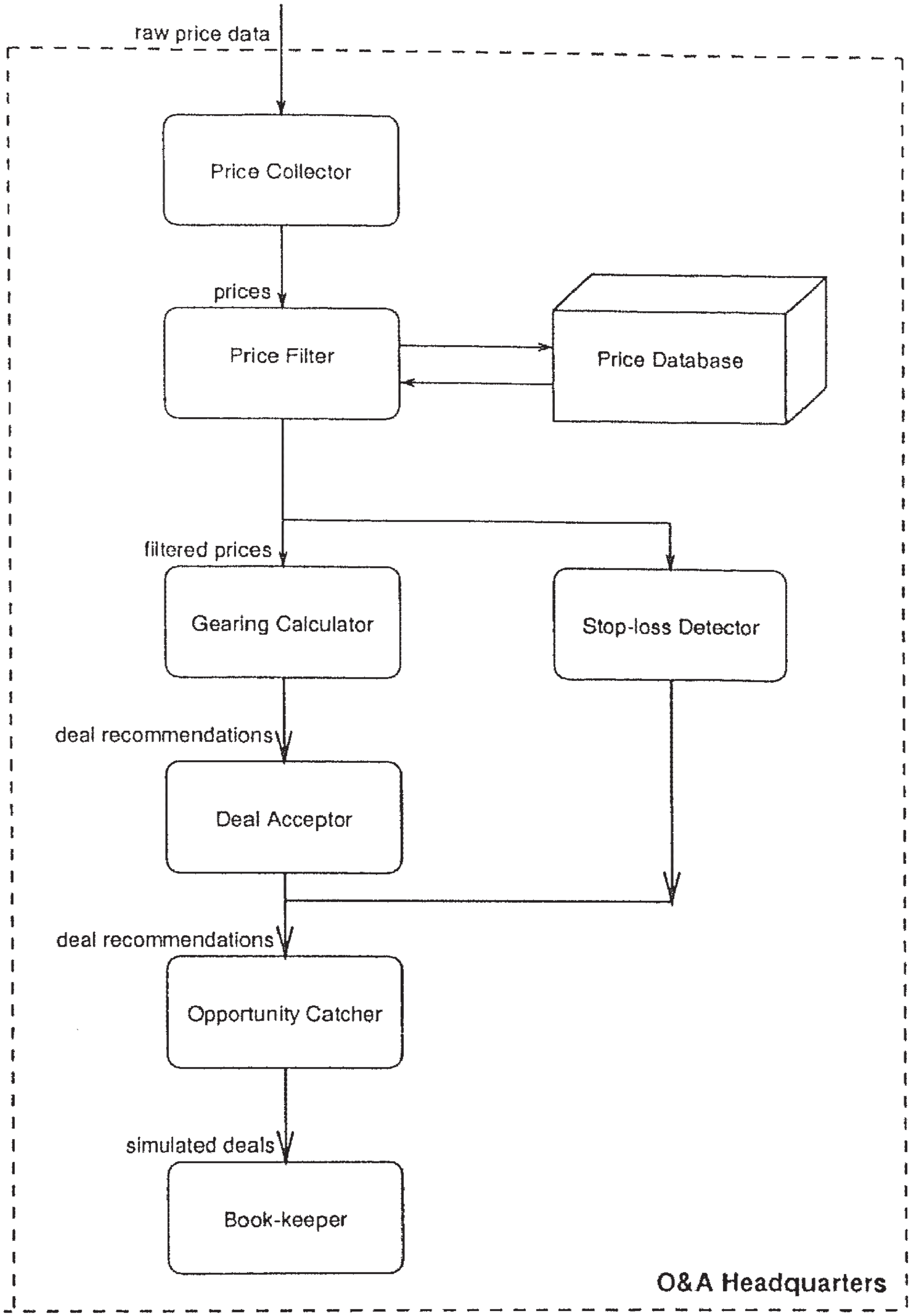

Fig. 2. *Data flow of prices and deal recommendations within a trading model*

trading positions by applying various rules. For instance, a model may enter a long position if an indicator exceeds a certain threshold. Such rules are discussed in more detail in section 3.2.

Other rules determine whether a deal may be made at all. Among various factors, these rules take the timing of the recommendation into consideration. Such authorization rules are discussed in section 3.3.

A complete trading model thus consists of a set of indicator computations combined with a collection of rules. The former are functions of the price history. The latter determine the applicability of the indicator computations in generating trading recommendations.

3.2. The gearing calculators

A gearing calculator lies at the heart of an O&A trading model. The gearing calculator provides the trading model with its intelligence and the ability to capitalize on movements in the FX markets. The gearing calculator also provides the trading model with particular properties. These include the frequency of dealing and the circumstances under which positions may be entered.

In other words, the gearing calculator is the *real* model. In contrast, the other trading model components form a shell around the gearing calculator, providing it with price data, detecting if the stop-loss is hit and examining the trading recommendations made by the gearing calculator. The gearing calculator reevaluates its position every time a new price datum is received from the quote-vendors. (As previously noted, a filter validates each price beforehand in order to eliminate outlyers and other implausible data).

The gearing calculator employs two kinds of software ingredients: a set of indicators which are produced from the input price data (discussed in detail in section 4) and trading rules which are functions of the past dealing history, the current position and other quantities such as the current unrealized return of an open position.

As the name suggests, an indicator provides a measure of whether a new position should be entered. These indicators are analyzed by the rules inside the gearing calculator to determine whether such a change of position should in fact take place. In the simplest form, an indicator crossing a predefined threshold may cause a rule to be activated that in turn causes such a change in position to occur. Thus the relative values of the indicators signify internal trading recommendations which are then refined through the application of various rules. Other, more complicated rules may modify the indicators' basic recommendations in additional ways:

1. By inhibiting a recommendation produced by the indicators. For example, if the price movements since the previous deal are too small in either direction, an indicator recommendation to reverse the present gearing is suppressed.

2. By changing the indicator thresholds themselves. While the model is in a neutral position, such a rule may increase the threshold values.

3. By choosing an opposite position to the one hinted at by the indicators. That is, by entering a "contrarian" position. (A contrarian position is long when prices are declining or short when prices are climbing).

4. By imposing new stop-loss values. A smaller stop-loss may be established if the recommended position is contrarian. The stop-loss is reduced because a contrarian strategy is considered risky.

The relative values of the indicators determine the gearing that should be used. That is, not only do the indicators determine the position to be entered (after subsequent transformation by the rules), they also determine the magnitude of the gearing to be used. As stated previously, the O&A models employ half-integral gearings in the range −1.0 to 1.0.

The other trading model components surrounding the gearing calculator do so through a standardized interface. We are therefore able to "drop" newly developed gearing calculators into the trading model environment. At the time of writing, we have produced three separate gearing calculators (designated classes 40, 50 and 60) each working with seven currencies (USD/DEM, USD/CHF, GBP/USD, USD/JPY, USD/ITL, USD/FRF and USD/NLG). These gearing calculators differ in their respective trading frequencies, risk profiles and other properties. Developing a new gearing calculator is thus akin to developing a new trading class while preserving the overall trading model structure. This allows us straightforwardly to customize existing gearing calculators or to experiment with new ones.

3.3. The deal acceptor

The fact that the gearing calculator's indicators and rules suggest entering a new position does not necessarily mean that the model will make such a recommendation. Whether it does or not depends on various secondary rules that then take effect.

These rules comprise the deal acceptor. This determines whether the deal proposed by the indicators is allowed to be made. The prime constraint is the timing of the proposed deal. First, no deal other than a stop-loss deal (see section 3.4.) may take place within fifteen minutes of a deal already having occurred. This is to prevent overloading a human dealer who may be following the models. Second, the gearing calculator may make a recommendation to enter a new trading position but this recommendation can be followed only if the local market is open.

The quality of the most recent price imposes another constraint. A stringent deal-filter determines if a given price is suitable for dealing. This is so that we can be sure that recommended deals are made only with genuine prices rather than extraneous data. The deal acceptor permits a new deal only with a price passing the deal-filter.

If the gearing calculator suggests entering a new position but the deal acceptor decrees otherwise, the suggestion is simply ignored. Eventually, when timing and other factors are right, the gearing calculator will suggest entering a new position and the deal acceptor will approve.

3.4. Stop-loss detection

Besides being passed on to the gearing calculator, the filtered price quotes are also sent to the stop-loss detector.

The stop-loss detector triggers if the market moves in the "wrong" direction. That is, if the model enters a trading position because it anticipates the market to move in a certain direction but in fact the market then moves the other way, the stop-loss may be hit.

The trading model defines a stop-loss price when a position is entered. If the current price — that is, the most recent price — moves below the stop-loss price (for a long position) or above the stop-loss price (for a short position), the stop-loss is said to be hit. Hitting the stop-loss causes a deal to a neutral position to be made. In effect, the stop-loss prevents excessive loss of capital should the market go the wrong way.

The stop-loss price may change when a new position is entered or as the current price changes (see section 3.5.). The current stop-loss price is displayed on the user-agent.

A stop-loss deal may occur at any time, even outside market hours. The assumption is that a position that is kept open outside market hours is handled by a colleague in the American or Far East markets who will deal appropriately if the stop-loss is hit. Should this happen, no further change in position occurs until the local market opens once again.

3.5. Stop-profit

The concept of stop-profit is associated with that of stop-loss. The stop-loss price starts to move in parallel with the current price once a trading model has achieved a potential profit (3% or slightly less) since entering the latest position. In other words, being in a situation whereby the model could realize such a gain by immediately entering a neutral position causes the stop-loss price to start moving. The difference between the stop-loss and current prices is kept constant as long as the current price continues moving in a direction that increases the potential profit of the open position. That is, the stop-loss price moves as a ratchet in parallel with the current price. The stop price is allowed to move only during opening hours. It is never adjusted when the market is closed.

The model then enters a neutral position if it detects prices slipping backwards. This allows a model to save any profit it has generated rather than lose it when the market abruptly turns. This one-directional movement of the stop-loss price allows the model to capitalize on a price trend.

3.6. The opportunity catcher

The trading model may thus make a deal recommendation in two distinct ways. One, the gearing calculator may make a recommendation that is then authorized by the deal acceptor. Two, hitting the stop-loss price activates the stop-loss detector.

Whichever way a deal comes about, the opportunity catcher is activated. The opportunity catcher manifests itself on the user-agent as an eye-catching signal for the FX dealer to buy or sell according to the recommendation.

While he is actively dealing, the opportunity catcher in the trading model collects the best price with which to deal, either the highest bid price if going from a longer position to a shorter one or the lowest ask price if going from a shorter position to a longer one. This search for the best price lasts for two or three minutes depending on the currency, the assumption being that a quoted price has a life-time of about two or three minutes even if it is superseded by later quotes.

After the two or three minute search period, a second signal appears on the user-agent signifying that the trading model has made a simulated deal using the best price found by the opportunity catcher. The FX dealer then concludes his deal-making activities and waits until the trading model produces another recommendation.

(As a point of detail, the opportunity catcher is not activated for a stop-loss deal occurring outside market hours. In this case the trading model deals directly. A human trader following the model should then make a corresponding deal for himself as quickly as possible.)

3.7. The book-keeper

The *book-keeper* makes simulated deals on behalf of the trading model. It keeps track of all deals that have been made and evaluates statistics demonstrating the performance of the trading model.

An important variable is the average price $\overline{p}$ paid for achieving the current gearing. After a new deal with index t, this quantity depends on the type of transaction as follows:

$$\overline{p}_t \equiv \begin{cases} \overline{p}_{t-1} & \text{if} \quad |g_t| < |g_{t-1}| \wedge g_t g_{t-1} > 0 \\ g_t \left[\frac{g_t - g_{t-1}}{p_t} + \frac{g_{t-1}}{\overline{p}_{t-1}} \right]^{-1} & \text{if} \quad |g_t| > |g_{t-1}| \wedge g_t g_{t-1} > 0 \\ p_t & \text{if} \quad g_t g_{t-1} < 0 \vee g_{t-1} = 0 \\ \text{undefined} & \text{if} \quad g_t = 0 \end{cases} \tag{3.1}$$

where g_{t-1} and g_t are the previous and current gearings respectively, p_t is the current transaction price, and $\overline{p}_{t-1}$ the average price before the deal. In the initial case, when the current gearing is neutral, the average price $\overline{p}$ is not yet defined.

The average price $\overline{p}$ is needed to compute a quantity central to a trading model, the *return* of a deal:

$$r_t \equiv (g_{t-1} - g_t)\left(\frac{p_t}{\overline{p}_{t-1}} - 1\right) \tag{3.2}$$

There are deals with no returns: those starting from a neutral gearing, $g_{t-1} = 0$, and those increasing the absolute value of the gearing while keeping its sign. In these cases, eq. 3.2 does not apply (whereas eq. 3.1 applies to all deals).

Both above equations allow the computation of a set of other quantities that are important for the different trading rules and for the performance evaluation of the models and which are computed in the book-keeper. These are the following:

- the *current-return* r_c. This is the unrealized return of a transaction when the current position is off the equilibrium ($g_t \neq 0$). If p_c is the current market price required for going back to neutral, generalizing eq. 3.2 yields the current-return,

$$r_c \equiv g_t\left(\frac{p_c}{\overline{p}_t} - 1\right) \tag{3.3}$$

- the *maximum return when open* is the maximum value of r_c from a transaction t to a transaction $t+1$ reached during opening hours,
- the *minimum return when open* is the minimum value of r_c from a transaction t to a transaction $t+1$ reached during opening hours,
- the *total return* is a measure of the overall success of a trading model over a period T and is simply equal to,

$$R_T \equiv \sum_{i=1}^{n} r_i \tag{3.4}$$

where n is the total number of transactions that generated a return during the period T. The total return expresses the amount of gains made by a trader who would always invest up to his initial capital or credit limit in his home currency.

- the *cumulated return* is another measure of the overall success of a trading model wherein the trader always reinvests up to his current capital including gains or losses,

$$C_T \equiv \prod_{i=1}^{n}(1 + r_i) - 1 \tag{3.5}$$

This quantity is slightly more erratic than the total return.

- the *maximum drawdown* D_T over a certain period $T = t_E - t_0$. It is defined as:

$$D_T \equiv \max(R_{t_a} - R_{t_b} | t_0 \le t_a \le t_b \le t_E) \tag{3.6}$$

where R_{t_a} and R_{t_b} are the total returns of the periods from t_0 to t_a and t_b respectively.

- the *profit over loss ratio* gives an idea of the type of strategy used by the model. Its definition is:

$$\frac{P_T}{L_T} \equiv \frac{N_T(r_i | r_i > 0)}{N_T(r_i | r_i < 0)} \tag{3.7}$$

where N_T is a function that gives the number of elements of a particular set of variables under certain conditions during a period T.

The book-keeper also computes other quantities but they are less relevant to this paper. Another book-keeper function is to save the trading history of a particular model so it can be retrieved at any time on a user-agent.

3.8. Status and simulation messages

Status and simulation messages are occasionally issued to keep the FX dealer informed of the trading models' states as well as of any likely deals.

Status messages typically give advance warning of the market opening or closing or notification if the market is already closed. These messages also indicate whether the supply of raw price data from the commercial market quoters to the O&A trading models has been disrupted for any reason. Ideally, of course, this should never happen but a slight possibility of some disruption outside the control of O&A does exist.

Simulation messages give advance notification of a probable deal. A deal is considered to be probable if the market continues its present trend for half an hour or so.

3.9. The G programming language

The trading models are event-driven. This means that each model is implemented in terms of defined actions that are triggered when some external event occurs. An external event might be the receipt of a new price from the commercial quote-vendors or it might be a timer event such as a local market opening or closing.

This event-driven characteristic of the trading models suggests an implementation in terms of a data-flow paradigm. Reception of a new price or a timer event causes some defined actions to be triggered and these, in turn, cause other actions to be triggered. An action may, for example, be a calculation with a new input price, generation of a new trading recommendation or the transfer of computed data to the user-agents. This process continues recursively, one action triggering others, until all dependent actions are executed.

The O&A trading models are therefore implemented in a computer language that directly supports such a data-flow paradigm. This is "G" [2], a specialized programming language developed at O&A for rule-based, real-time data analysis.

The name G stands for glue because the G interpreter glues together disparate software components into a unified whole. In common with conventional programming languages, G provides syntactic constructs for defining variables, expressions and functions. It also include a data-driven knowledge-based component for problem solving. G provides means for data to be received from external sources such as a price stream or timer input. Other connections to the real world (file input and output, message-passing) are also supported.

As noted above, G is a data flow language. The syntax is taken from the LISP programming language. Newcomers to G are often surprised by the preponderance of defined expressions and the relative lack of the more usual programming control constructs such as procedure calls, `while` and `for` loops. These do exist in G but their usage is infrequent. Rather the intention is that one or more input streams to G provide input in the form of variables being updated. Actions and expressions defined in terms of these variables are updated in turn and this process continues recursively until all dependent expressions are updated. The order in which expressions are evaluated may be precisely defined for fine control of this "trickle down" effect. Programming in G, therefore, comes down to defining expressions and their mutual dependence and this has quite a different flavor from programming in better-known languages.

For example, the following G fragment defines two variables, a and b, and two expressions, e1 and e2. Expression e1 is written in terms of a and b and expression e2 is written in terms of e1. Subsequently setting either variable, a or b, to a new value causes both e1 and e2 to be automatically reevaluated:

```
(defvar a (real 1.0)          ; define variables a and b
        b (real 2.0))         ; ...  with initial values
(defexpr e1 (+ a b)           ; define expr e1 in terms of a and b
         e2 (* e1 7))         ; define expr e2 in terms of e1
(print a b e1 e2)             ; look at these values
1.000000 2.000000 3.000000 21.000000
(set a 8.0)                   ; give a new value to a
(update)                      ; evaluate dependent expressions
(print a b e1 e2)             ; look at these values once more
8.000000 2.000000 10.00000 70.00000
```

Other than the differences noted above, however, G shares many features with conventional programming languages. Strong typing, debugging support (break-pointing and tracing) and execution profiling are present in G.

3.10. Rules and meta-rules

An input stream to a G program provides price data and other input in the form of variables being updated. (Such variables are defined using language constructs that establish the connection between the input stream and the variables in question). Dependent expressions are then recursively evaluated before new price data is subsequently provided. However it is not possible to change the default order of evaluation within a set of G expressions which is that expressions are evaluated in their order of definition.

For developing realistic models we need other language constructs allowing us to prevent confusion between the different clusters of knowledge which can come into operation as circumstances vary during trading model operation. These are implemented in G through the ability to define rules with varying priorities and hence to change dynamically the order of their evaluation.

A G rule definition specifies a condition to be matched, an action to be executed and a priority, given as an integer value. For example:

```
(defrule rulename  priority  condition  action)
```

A rule is invoked if a variable used in the `condition` part is updated. Then, while the `condition` is fulfilled and the rule has the highest priority of all those whose respective `conditions` are also true, the `action` is executed. Priorities may change during execution of a program and hence the relative precedences of the rules vary correspondingly. A rule with a negative priority is disabled altogether.

In a trading model environment, rules provide a way of ensuring that the separate components such as the opportunity catcher, deal-acceptor, simulation and so on do not interfere with one another. For example, simulation is disabled while the opportunity catcher is running. Such separation of activities is implemented in terms of meta-rules. These are rules that monitor the activity of the other rules and expressions within the trading model code. Depending on what is happening inside the trading models, the meta-rules set priorities of other rules thereby enabling or disabling other components. In short, inclusion of meta-rules prevents the various trading model components from chaotically disrupting one another. For example:

```
;;;
;;; Activate/deactivate dealing.
;;;
(defrule
    deactivate-deal-execution      metarule-priority
        (| building-up? sim-active? opp-update?)
        ;;; then
        (set priority.deal-execution deactivate-priority)
)
...
(defrule
    deal-acceptor    priority.deal-execution
    ...
)
```

The meta-rule `deactivate-deal-execution` sets the priority of the rule `deal-acceptor` if the specified condition becomes `true`. This happens when the model is building up its moving averages before they are completely initialized (`building-up?`), when the model is performing a simulation (`sim-active?`) or if the model has activated the opportunity catcher and is choosing a price for making a transaction (`opp-update?`). A second meta-rule, not shown here, can later reactivate the `deal-acceptor` rule if another condition is subsequently fulfilled.

4. The Model Indicators

As noted above, the indicators form a pivotal component of the gearing calculator and hence of the dealing decision process. An indicator is a function only of time and the price history. It summarizes relevant information of the past price movements in the form of a single variable.

4.1. The modified business time scale

The indicator definition needs a time scale on which the price history is analyzed. The usual physical time scale t (as used in physics) is inappropriate, as we have demonstrated in many tests. For example, the market perceives a price history differently when it covers two working days in one case and two weekend days (Saturday and Sunday) in another, even if the price curves against t have the same form in both cases. A *business* time scale, related to business activity, is a better choice for analyzing price history.

Changing the time scale can be viewed as a way to introduce some of the "fundamentals" that are missing in conventional time series analysis. These fundamental economic variables certainly have an influence on price movements, but their individual and combined effects are difficult to isolate and thus to replicate. Their impact can be better seen, however, if one views price movements as indelible "footprints" left by these variables. A time scale based on price movements may therefore be used to capture the tracks left by economic factors.

Conventional business time scales are quite simple: they omit inactive periods such as weekends and count only the business hours. A subtler "modified business time" scale is achieved by expanding periods of high activity and contracting but not omitting periods of low activity.

The 24 hour FX market indeed has periods of low activity (for example, the noon break in East Asia) and high activity (for example, when the European and American business hours overlap), see [3]). In [1], a modified business time scale called ϑ-time is introduced modeling the daily and weekly fluctuations of activity. This ϑ-time is adopted here as the time scale for indicator computations.

The ϑ-time model considers business activity in terms of statistical means of absolute price movements (a volatility measure). Hourly or daily FX transaction volume figures would be another alternative for measuring business activity, but such data is unavailable. The discussion in [1] argues for using price movements in a business time scale.

The business activity exhibits distinct daily and weekly seasonal heteroskedasticity which can be attributed to the changing presence of traders on the FX markets. The ϑ-time model defines the activity $a(t)$ as the sum of three activity functions $a_k(t)$, corresponding to three sub-markets of the worldwide FX market which are geographically centered in East Asia, Europe and North America. More precisely, the $a_k(t)$ are functions of the time within a day and of the nature of the weekday (working day or weekend day). The ϑ-time is the integral of $a(t)$ over physical time t.

The three sub-markets model succeeds in explaining the observed seasonal heteroskedasticity: hourly absolute price changes lose their seasonality when analyzed against ϑ-time instead of t.

In addition to the daily and weekly seasonal heteroskedasticity, FX prices also exhibit non-seasonal heteroskedasticity in form of clusters of high or low volatility, as found by many authors in recent years. The ϑ-time model does not attempt to take these into account. Therefore, ϑ has a regular pattern with a period of 1 week and may also be determined for the future. It is calibrated so that one full week in ϑ-time corresponds to one week in physical time t. Hence it can be measured in the same units as t; for example, hours, days, weeks or years. Details on the modified business time ϑ can be found in [1].

4.2. The EMA operator and its repeated application

A trading model indicator is defined as a momentum, the difference of the current logarithmic middle price x and its moving average (MA) computed in ϑ-time:

$$m_x(\Delta\vartheta_r, \vartheta) \equiv x - MA_x(\Delta\vartheta_r, \vartheta) \tag{4.1}$$

where $\Delta\vartheta_r$ is the MA range, a measure related to the depth of the past considered in the MA. The moving average is a useful tool for summarizing the past behavior of a time series at any time point. (An alternative definition of a momentum would be the difference of two moving averages with different ranges. We do not use this because the definition of eq. 4.1 has always proved superior in our trading model tests).

A special moving average operator is used which we describe here. Among the MAs, the exponentially weighted moving average (EMA) plays an important role. Its weighting function declines exponentially with the time distance of past observations from the present. Sequential EMAs along a time series are straightforward to calculate with the help of a recursion formula; this is more efficient than the computation of differently weighted MAs.

Although moving averages are well known for homogeneous time series[3], their formalism is much less developed for the case of continuous time series in which we are interested. Our data is not equally spaced over time (neither t nor ϑ), but can be interpolated in the time intervals between the price quotes. We choose linear interpolation in ϑ-time as our interpolation method. In some alternative tests, we have also taken the previous quote as a function value for the whole interval. Through interpolation, we arrive at a continuous time function $x(\vartheta)$, for which the EMA is an integral,

$$EMA_x(\Delta\vartheta_r, \vartheta_c) \equiv \frac{1}{\Delta\vartheta_r} \int_{-\infty}^{\vartheta_c} e^{\frac{\vartheta_c - \vartheta}{\Delta\vartheta_r}} x(\vartheta) d\vartheta \tag{4.2}$$

where c is the index of the current time series element and the range $\Delta\vartheta_r$ is the center of gravity of the EMA weighting function. This EMA can be computed efficiently with a recursion formula:

[3]For instance, see Granger and Newbold (1977) [4].

$$'MA_x(\Delta\vartheta_r, \vartheta_c) = \mu EMA_x(\Delta\vartheta_r, \vartheta_{c-1}) + (1-\mu)x_c + (\mu - \nu)\Delta x_c, \quad (4.3)$$

-e Δx_c is defined as $x_c - x_{c-1}$. For a time series representing a function all types of interpolation, we obtain

$$\mu = e^{-\alpha} = e^{-\frac{\Delta\vartheta_c}{\Delta\vartheta_r}}, \quad (4.4)$$

-e the time interval $\Delta\vartheta_c$ is defined as $\vartheta_c - \vartheta_{c-1}$, and

$$\nu = \frac{1-e^{-\alpha}}{\alpha} = \frac{1-\mu}{\alpha} \quad (4.5)$$

[his is valid under the assumption of linear interpolation between subse-ıt series elements. If the function value within such an interval is assumed e constant and equal to the preceding series element, then the recursion ıula also applies by setting $\nu = 1$ instead of eq. 4.5.

[ogether with the recursion formulae, the initial value EMA value must be specified. There is usually no information before the first series element which thus becomes the natural choice for the initialization:

$$EMA_x(\Delta\vartheta_r, \vartheta_1) = x_1. \quad (4.6)$$

[he error made by this initialization declines by a factor $e^{-(\vartheta_c - \vartheta_1)/\Delta\vartheta_r}$. art of the data must therefore be reserved to build up the EMAs thus ıitting this error to become small.

[he exponential weighting function with its steeply increasing form gives ong weight to the recent past which may contain noisy short-term price ement structures. These can lead to inaccurate trading signals. For this on, our tests have found other, less peaked MA weighting functions to be e successful.

Ve attain different MA weighting functions by repeated application of the A operator. The EMA operator yields a result of the same mathematical re as its input: a time series. Because of this property, the EMA oper- can be applied iteratively. The result at each stage is a moving average) with a weighting function more complicated than a simple exponential tion:

$$EMA_x^{(n)} \equiv EMA(EMA_x^{(n-1)}(\Delta\vartheta_r, \vartheta_c)) \quad (4.7)$$

-e $EMA_x^{(0)}(\Delta\vartheta_r, \vartheta_c) \equiv x_c$. The order of the EMA operator (how many s the simple EMA operator is applied) determines the kind of weighting ied to the past prices.

n our most successful trading models, the moving average is the mean of As of different orders,

$$EMA_x^{(m,n)} \equiv \frac{1}{n-m+1} \sum_{i=m}^{n} EMA_x^{(i)}(\Delta\vartheta_r, \vartheta) \tag{4.8}$$

This MA has a weighting function with the center of gravity at $(m+n)\Delta\vartheta_r/2$. The weighting function has the following properties:

- a low weight for the most recent past,
- a wide, plateau-like weight maximum at time points around the center of gravity (no sharp peak),
- a rapidly but smoothly declining weight for more distant times.

The indicator is a momentum in the sense of eq. 4.1:

$$I_x = \frac{1}{s(\Delta\vartheta_r, m, n)}(x - EMA_x^{(m,n)}) \tag{4.9}$$

where $s(\Delta\vartheta_r, m, n)$ is a scaling factor.

4.3. Threshold values for trend and overbought/oversold signals

Trading signals are given when the indicator I_x crosses over certain threshold values. There are two types of trading signals:

- Trend signals: these are recommendations to follow the trend indicated by I_x. Good threshold values are somewhat low so that trends can be detected relatively early. A "buy" signal is given when I_x exceeds such a value, a "sell" signal when I_x exceeds the corresponding negative value.
- Overbought/oversold signals: these are contrarian instructions to stop following an old trend or to take the opposite position. Good threshold values for these signals are high, at levels that are only exceptionally reached by I_x. "Overbought" refers to high positive values, "oversold" to high negative ones.

Our trading models use several threshold values of both types. Different threshold values are used in different situations depending on the current position of the trading model and other internal variables of the decision-making process. Some trading signals recommend taking a new position with full gearing, others a new position with only half gearing, again depending on certain internal conditions of the decision-making process. In many cases, the decision-making process does not follow a trading signal because another decision rule has precedence, as discussed in in section 3.10.

5. Optimization and Testing Procedures

5.1. Historical testing

The O&A trading models are designed for use in a real-time service. However, for testing purposes, for tuning the trading models and for producing statistics, we need a more stable, reproducible environment. Thus, we are able to drive our trading models from a database of historical price data. That is, the defined actions are triggered as though they were being fed from a real-time stream. The trick here is to use the historical price database to form a stream of combined price and timer events. In other words, the timer events are simulated from the historical prices.

This interchangeability between historical and real-time modes is invaluable for the trading model developer. Once a new gearing calculator has been produced, for example, the developer can test it using historical price data. Without making any code changes, he can then switch the trading model into real-time mode so that it then reacts to new price data as they become available.

5.2. Optimization and definition of trading model performance

Optimizing trading models with minimum overfitting is a difficult task. Overfitting means building the indicators to fit a set of past data so well that they are no longer of general value: instead of modeling the principles underlying the price movement, they model the specific price moves observed during a particular time period. Such a model usually exhibits a totally different behavior or may fail to perform out-of-sample; that is, with data that was not used for the optimization process. To avoid overfitting during optimization, we need a good measure of the trading model performance and to use robust optimization and testing procedures. A strict division of the available historical data used in the different phases of optimization and testing is also important.

The first step is to define a value describing the trading model performance in order to minimize the risk of overfitting in the in-sample period and also to be able to compare different trading models against one other. The performance of a trading model cannot be judged by its total return only; the relevant issue is the overall risk profile. We need a measure of the trading model performance that accounts for the set of following requirements:

- the best total return,
- a smooth, almost linear increase of the total return over time,
- a small clustering of losses,
- no bias towards low frequency trading models.

A measure frequently used to evaluate portfolio models is the Sharpe index. Unfortunately, this measure neglects the risk due to unrealized losses while the model stays in one position. It also introduces a bias in favor of models with a low dealing frequency. Furthermore the Sharpe index is numerically unstable for small variance of returns and cannot consider the clustering of profit and loss trades. This measure clearly does not fulfill our requirements. In the next section, we present a measure of the trading model performance which is risk-sensitive and indicates the stability of returns.

5.3. Risk-sensitive measure of the trading return

As a basis of our risk-sensitive performance measure, a new trading model return variable $\tilde{R}$ is defined to be the sum of the total return R (eq. 3.4) and the non-realized current return r_c (eq. 3.3). The variable $\tilde{R}$ is more continuous over time than R and reflects the additional risk due to non-realized returns. Its change over a test time interval Δt is termed $X_{\Delta t}$:

$$X_{\Delta t} = \tilde{R}(t) - \tilde{R}(t - \Delta t), \qquad \text{where} \qquad \tilde{R}(t) = R(t) + r(t) \tag{5.1}$$

We can make N independent observations of $X_{\Delta t}$ within a total test period ΔT, where

$$\Delta t = \frac{\Delta T}{N} \tag{5.2}$$

A risk-sensitive performance measure of the trading model comparable to the average return can be deduced from the utility function formalism of Keeney and Raiffa [5]. We assume that the variable $X_{\Delta t}$ is stochastic and follows a Gaussian random walk about its mean value $\overline{X}_{\Delta t}$ and that the risk aversion C is constant with respect to $X_{\Delta t}$. The resulting utility $u(X_{\Delta t})$ of an observation is $-\exp(-CX_{\Delta t})$, with an expectation value of $\overline{u} = u(\overline{X}_{\Delta t})\exp(-C^2\sigma^2_{\Delta t}/2)$, where $\sigma^2_{\Delta t}$ is the variance of $X_{\Delta t}$. This expectation of the utility can be transformed back to a variable termed the *effective return*: $X_{eff,\Delta t} = -\log(-\overline{u})/C$. The following definition is obtained:

$$X_{eff,\Delta t} = \overline{X}_{\Delta t} - \frac{C\sigma^2_{\Delta t}}{2} \tag{5.3}$$

The risk term $C\sigma^2_{\Delta t}/2$ can be regarded as a risk premium deducted from the original return. The variance is computed from the sample:

$$\sigma^2_{\Delta t} = \frac{N}{N-1}\left(\overline{X^2_{\Delta t}} - \overline{X}^2_{\Delta t}\right) \tag{5.4}$$

For a particular time horizon Δt, the variable $X_{eff,\Delta t}$ is our new measure of trading model performance including risk. Unlike to the Sharpe index, this measure is numerically stable and can differentiate between two trading models with a straight line behavior ($\sigma^2_{\Delta t} = 0$) by choosing the one with the best average return. Moreover, the definition given in eq. 5.3 gives the bookkeeping details of a trade a smaller role than the Sharpe index because $X_{\Delta t}$ also contains elements of the current return and is computed at regular time intervals Δt.

The measure $X_{eff,\Delta t}$ still depends on the size of the time interval Δt. It is hard to compare $X_{eff,\Delta t}$ values for different intervals. The usual way to enable comparisons between different intervals is through annualization: multiplication by the annualization factor, $A_{\Delta t} = 1\text{year}/\Delta t$,

$$X_{eff,ann,\Delta t} = A_{\Delta t} X_{eff,\Delta t} = \overline{X} - \frac{C}{2} A_{\Delta t} \sigma^2_{\Delta t} \tag{5.5}$$

where $\overline{X}$ is the annualized return, no longer dependent on Δt. In the second term of the last form of eq. 5.5, we find the factor $A_{\Delta t}\sigma^2_{\Delta t}$. This factor has a constant expectation, independent of Δt, if the return function $\tilde{R}(t)$ is assumed to follow a Gaussian random walk in addition to a linear drift. For such a return function (our reference case), we introduce the condition that the expectation of $X_{eff,ann,\Delta t}$ must not depend on Δt. This condition is fulfilled only if the risk aversion C is constant, that is, independent of Δt. Annualized effective returns $X_{eff,ann}$, computed for different intervals Δt by eq. 5.5 with a constant C value, can therefore be directly compared.

This measure, though annualized by eq. 5.5, still has a risk term associated with Δt and is insensitive to changes occurring with much longer or much shorter horizons. To achieve a measure that simultaneously considers a wide range of horizons, we introduce a weighted average of several $X_{eff,ann}$ computed with n different time horizons Δt_i, and thus take advantage of the fact that annualized $X_{eff,ann}$ can be directly compared:

$$X_{eff} = \frac{\sum_{i=1}^{n} w_i X_{eff,ann,\Delta t_i}}{\sum_{i=1}^{n} w_i} \tag{5.6}$$

where the weights w_i can be chosen according to the relative importance of the time horizons Δt_i and may differ for trading models with different trading frequencies. Substituting $X_{eff,ann}$ by its expression (eq. 5.5), X_{eff} becomes

$$X_{eff} = \overline{X} - \frac{C}{2} \frac{\sum_{i=1}^{n} w_i \frac{A_i\ \sigma_i^2}{\Delta t_i}}{\sum_{i=1}^{n} w_i} \tag{5.7}$$

where the variance σ_i^2 is computed with eq. 5.4 for the time horizon Δt_i, and A_i is the corresponding annualization factor ($= 1\text{year}/\Delta t_i$). Because $\sigma_i^2 \geq 0$, we have $X_{eff} \leq \overline{X}$. By empirically balancing risk and return of some test trading models, we found values between 0.08 and 0.15 to be reasonable for C.

By adopting this new measure, we depart from the formal utility function theory defined by Keeney and Raiffa [5]. This theory is based on the additivity of utilities, but in eq. 5.6, we average effective returns (which are non-linear functions of utilities). Nevertheless, we choose this definition because we do not see the utility of each horizon as a component of a meta-utility but rather as representing a typical segment of the market. If one of these segments endures a bad phase, its influence on the overall outcome need not be over-proportional which would be the case if we kept the formalism of additive utilities.

In the discussion of eq. 5.5, we have shown that the risk aversion C has no systematic dependence on the horizon Δt_i. However, dealers using a trading model might perceive differently the risks of various horizons. We might introduce special C values for individual horizons according to their trading preferences. However, we can achieve an equivalent effect by changing the weights w_i, which are already differentiated for each horizon. These weights reflect the importance of the horizons in terms of the risk sensitivity associated with each horizon.

Specifically, in our performance measure, we use a standard weighting function that determines the weights w_i and thus the relative importance of the different horizons.

$$w(\Delta t) = \frac{1}{2 + \left(\log \frac{\Delta t}{90\text{days}}\right)^2} \tag{5.8}$$

The weight maximum is set to the 90 days horizon in order to give sufficient importance to the short horizons compared with the long ones. This weighting function is designed to be applied to horizons Δt_i in a roughly geometric sequence.

An approximately geometric sequence of n horizons Δt_i is chosen with the following simple construction. Once a testing period (full sample of size ΔT) has been established, it is divided by 4. If this division results in a time horizon longer than 2 years, then the result is divided by 2 and so on, until a horizon Δt_1 strictly shorter than 2 years is reached. We limit this highest horizon because dealers usually close their books after one year and are less sensitive to return clustering on longer horizons. The next horizon is obtained by a division by 2 of the previous one and so on until a last horizon between 5 to 10 days is reached; this shortest horizon is then forced to be $\Delta t_n =$ 7 days. All horizons are truncated to full days. If there is no integer multiple of a resulting Δt_i that exactly covers the full sample, then the first analyzed interval at the start of the full sample is extended accordingly. The exact Δt_i values to be inserted in eq. 5.7 and eq. 5.8 are the results of eq. 5.2.

Δt_i	594	297	148	74	37	18	7
w_i	0.086	0.139	0.212	0.234	0.171	0.104	0.056
σ_i	21.92%	11.80%	7.81%	5.37%	3.64%	2.41%	1.47%
$\overline{X}_{\Delta t_i}$	36.83%	18.42%	9.21%	4.60%	2.30%	1.12%	0.43%
$X_{eff,ann,\Delta t_i}$	7.88%	14.09%	15.15%	15.56%	16.12%	16.75%	16.99%

Tab. 1. *Typical results for the performance measure according to each horizon. The horizons Δt_i are given in days, the weights are normalized to one.*

To illustrate the effective return computation, we show in Tab. 1. typical results for a trading model for the German mark against the US dollar tested on six and a half years (March 1986 to September 1992) of data. The analyzed horizons Δt_i, the weights w_i, the variances σ_i^2, the average returns $\overline{X}_{\Delta t_i}$, and the annualized effective returns $X_{eff,ann,\Delta t_i}$ (see eq. 5.5) are presented in this table. The average yearly return of this run is $\overline{X} = 22.65\%$ and the effective yearly return (computed according to eq. 5.7 with $C = 0.10$) is $X_{eff} = 14.91\%$. The yearly return is reduced by a "risk premium" of about a third of the original value.

5.4. Model optimizations

In order to optimize and test our trading models, we split the available historical data into three different periods. The first period is used to build-up the indicators, the second one is for the optimization of the trading model parameters and the third for selecting the best trading models. The continuously increasing data set collected from the quote-vendors is reserved for real-time (*ex ante*) testing.

The build-up period generally contains more than ten years of daily data which is used to update long term indicators. A few weeks of tick-by-tick data is also needed to build up short term indicators. The end of that period is 1 March 1986 for the USD/DEM, USD/JPY, GBP/USD and USD/CHF major exchange rates and 1 December 1986 for the USD/NLG, USD/FRF and USD/ITL minor exchange rates.

To optimize the trading model parameters, we use a three years in-sample period starting just after the end of the build-up period. We assume that three years of tick-by-tick data is sufficient to optimize our trading models. The end of the latter period is 1 March 1989 for the major exchange rates and 1 December 1989 for the minor ones.

The in-sample performance refers to the period for which the model is developed and optimized. Different formulae for computing the indicators, different trading strategies and different parameters are tested until the model is found to achieve the optimal return. This process involves thousands of simulation runs. To select the best parameter set, we choose the solution in the parameter space that corresponds to the larger effective return. If different solutions with comparable returns are found, we choose the solution which is not too sensitive to a small change of the parameter set. In some cases, a period of three years in-sample data is insufficient because some volatility clusters can be as large as the optimization period.

Because extending the available historical tick-by-tick data is not possible, we enforce further restrictions on the choice of possible parameters. A solution is considered only if a similar cluster of good results exists for different FX rates at about the same place in the parameter space. This implies that various parameters must be scaled with the volatility of the individual FX rates in order to make them comparable between different rates.

The result of the in-sample optimization process is a small set of diverse trading models sharing a high effective return. Only these models are analyzed in the out-of-sample performance test. The out-of-sample period covers the remainder of the historical data, from the end of the in-sample period until the end of September 1991. A trading model is considered to be valid if the in-sample and out-of-sample periods yield a similar quality. Otherwise, the model is rejected. To avoid overfitting problems, the out-of-sample results are never used to select between similar models.

Since these models were developed we can now test them on a significant *ex ante* period. (We term *ex ante* period a period of time that has never been used in conjunction with model development). Our models have been running real-time since October 1991 so we have approximately one year of *ex ante* results.

6. Performance Analysis

A private investor may use a margin account to invest a multiple of his equity in the market. Such a leverage factor has an overriding impact on the total return. But the higher the leverage the higher too the risk of large drawdowns. Working with a high leverage factor may lead to margin calls during periods of drawdown, thus jeopardizing an investor's trading strategy. We have therefore developed our models without leverage. This should be borne in mind while evaluating the results.

6.1. Performance measures

The most important performance measure is the effective return as defined in eq. 5.7. While the total return may mask a considerable risk introduced by high volatility of return, the effective return is risk-sensitive: the higher the volatility of return the lower the effective return. In other words, high effective returns indicate highly stable returns.

The maximum drawdown defined by eq. 3.6 is also a significant performance indicator for a trading model user. It is the largest loss from a local maximum to a minimum of the total return curve within the test period. The maximum drawdown is a measure of risk, similar to the difference between the total return and the effective return.

Another important performance measure is the profit/loss ratio, defined in eq. 3.7 to be the ratio between the total number of profitable deals and the total number of losing deals. The present O&A models let profits run and cut losses early; they are almost always in an open position and go neutral only in case of stop-loss or save profit. Such a strategy is bound to result in a lower profit/loss ratio than would be achieved by less active models.

6.2. In-sample, out-of-sample and ex ante performance results

The first quality test of a new trading model is to compare the in-sample and out-of-sample performances. The performances in the two periods must be of the same quality for it to merit further consideration. To receive sufficient statistical information, a model must be tested for an extended out-of-sample period and for many different time series. If the average out-of-sample performance is significantly smaller it may mean that the model was overfitted. However, a much lower average volatility in the out-of-sample period can explain a reduced performance of the model in that there are fewer opportunities for profitable trading. The performance difference, however, must not be too large.

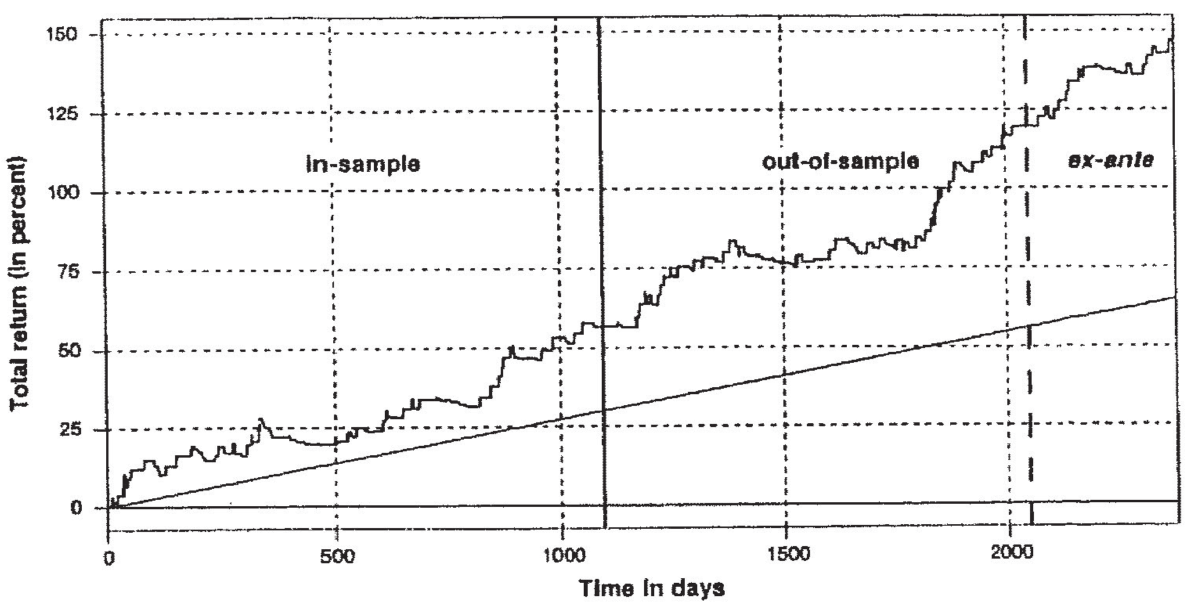

Fig. 3. *Example of total return against time for USD/DEM from March 1986 to September 1992. The straight line represents a total return of 10% per year. The different test periods are marked.*

An example of a total return curve over all three periods, in-sample, out-of-sample, and *ex ante*, is given in Fig. 3. The results of Tab. 1 belong to the same example.

FX rate	Sample	$\overline{X}$	X_{eff}	D	P/L
USD/DEM	in sample	19.1%	13.5%	8.4%	0.48
	out of sample	26.9%	19.0%	7.6%	0.41
	ex ante	28.8%	22.2%	3.9%	0.69
USD/JPY	in sample	15.1%	9.7%	7.5%	0.61
	out of sample	13.4%	8.7%	8.2%	0.70
	ex ante	-0.2%	-1.1%	4.2%	0.57
GBP/USD	in sample	14.6%	8.9%	11.1%	0.43
	out of sample	20.9%	14.9%	7.5%	0.46
	ex ante	13.7%	11.8%	7.1%	0.38
USD/CHF	in sample	18.4%	14.4%	6.9%	0.69
	out of sample	12.5%	6.2%	12.0%	0.51
	ex ante	22.7%	15.7%	3.2%	0.68
USD/FRF	in sample	19.3%	13.0%	10.3%	0.48
	out of sample	17.4%	12.4%	6.3%	0.48
	ex ante	9.1%	6.5%	5.7%	0.49
USD/NLG	in sample	20.3%	16.6%	7.6%	0.63
	out of sample	19.6%	11.1%	8.4%	0.53
	ex ante	28.9%	20.4%	4.2%	0.74
USD/ITL	in sample	11.2%	5.8%	13.3%	0.46
	out of sample	20.2%	14.9%	6.4%	0.47
	ex ante	18.8%	13.5%	2.9%	0.56
Average	in sample	16.9%	11.7%	9.3%	0.54
	out of sample	18.7%	12.5%	8.1%	0.51
	ex ante	17.4%	12.7%	4.5%	0.59

Tab. 2. *The in-sample period (1.3.86 to 1.3.89 for USD/DEM, USD/JPY, GBP/USD and USD/CHF, and 1.12.86 to 12.3.89 for USD/NLG, USD/FRF and USD/ITL), the out-of-sample period (1.3.89 to 1.10.91 and 1.12.89 to 1.10.91 respectively) and the* ex ante *period (1.10.91 to 4.9.92). The performance indicators shown here are: the annualized return ($\overline{X}$), the risk-sensitive return (X_{eff}), the maximum drawdown (D) over the sample, and the ratio of the number of profitable deals to the number of deals that resulted in a loss (P/L).*

In Tab. 2. we compare the in-sample, out-of-sample and *ex ante* performances of the O&A trading models. We observe that the average values of these quantities are similar to one another. This indicates the overall stability of our trading models and that the achieved profits do not result from model overfitting.

The individual *ex ante* values do, however, vary. We attribute this variance to the relative short *ex ante* testing period. The USD/JPY model has also performed less adequately than the others. This is because of the relatively low volatility in USD/JPY trading over the *ex ante* testing period.

In Tab. 3., the performance of the O&A class 40 trading models is compared to that of a more conventional 20-day moving-average model. Both models share the same environment; they have the same opening hours, bookkeeping rules and so on. They differ in the indicators and the rules on how to use these. The 20-day moving-average model uses the difference between the current logarithmic middle price and a conventional 20-day moving-average as its indicator, rather than eq. 4.9. In this simple model, the indicator threshold level is not varied and there are no overbought/oversold signals.

The O&A class 40 trading models surpass the 20-day moving-average model in all respects: they produce much higher total returns and effective returns and, except for one rate, much lower drawdowns.

FX rate	Model	$\overline{X}$	X_{eff}	D	P/L
USD/DEM	MA(20)	8.8%	3.3%	14.3%	0.37
	O&A model 40	*21.9%*	*14.1%*	*7.7%*	*0.41*
USD/JPY	MA(20)	−0.7%	−7.1%	39.9%	0.16
	O&A model 40	*16.2%*	*12.1%*	*8.2%*	*0.66*
GBP/USD	MA(20)	8.7%	4.3%	9.3%	0.45
	O&A model 40	*17.7%*	*11.4%*	*10.9%*	*0.43*
USD/CHF	MA(20)	7.9%	1.1%	18.9%	0.35
	O&A model 40	*17.7%*	*10.7%*	*13.0%*	*0.62*
USD/NLG	MA(20)	10.5%	6.3%	12.8%	0.41
	O&A model 40	*20.1%*	*14.2%*	*8.9%*	*0.57*
USD/FRF	MA(20)	8.8%	4.4%	12.7%	0.30
	O&A model 40	*22.1%*	*15.9%*	*8.6%*	*0.49*
USD/ITL	MA(20)	6.6%	2.3%	14.4%	0.34
	O&A model 40	*15.3%*	*10.3%*	*9.2%*	*0.46*

Tab.3. *Between the O&A class 40 trading models and a more conventional 20-day moving-average model. The test was conducted from 3.3.86 to 1.10.91 (major rates) and from 3.12.86 to 1.10.91 (minor rates). The same performance indicators as in Tab. 2. are displayed.*

6.3. Trading model portfolios

When investing in equity, it is common practice to spread the risk over different stocks. The same strategy may also be applied for currency trading. For example, the volatility may suddenly decline in a particular market and it may then not be possible to make any real money in this market for some period of time. We have tested such portfolios with a stable and equal capital distribution among the different models. The results (see Tab. 4) show clearly that the maximum drawdown and the drawdown period are significantly lower than those of the individual models. In line with portfolio theory, the portfolios with the lowest values (a little above 5% for the maximum drawdown and 22 (21) days for the drawdown period) are the ones comprising the most diversified currencies. At the same time these portfolios generate an impressive annualized return. Compared to the individual models, therefore, a portfolio achieves a more stable return with lower risk.

	O&A Trading Model			O&A Trading Model Portfolios		
Exchange rates	Annualized return	Maximum drawdown	Drawdown period	Annualized return	Maximum drawdown	Drawdown period
USD/GBP USD/DEM USD/JPY	17.61% 23.06% 14.19%	11.07% 8.36% 8.24%	95 days 133 days 21 days	18.35%	5.24%	22 days
USD/DEM USD/CHF USD/JPY	23.06% 15.77% 14.19%	8.36% 12.00% 8.24%	133 days 142 days 21 days	17.73%	5.41%	22 days
USD/GBP USD/CHF USD/JPY	17.61% 15.77% 14.19%	11.07% 12.00% 8.24%	95 days 142 days 21 days	15.95%	6.26%	21 days
USD/DEM USD/CHF USD/GBP	23.06% 15.77% 17.61%	8.36% 12.00% 11.07%	133 days 142 days 95 days	18.95%	6.62%	149 days
USD/FRF USD/DEM USD/JPY	18.51% 23.06% 14.19%	10.28% 8.36% 8.24%	129 days 133 days 21 days	18.28%	6.54%	129 days

Tab. 4. *Portfolios of three class 40 trading models with a stable, equal distribution of capital among the three models over a test period from 3 March 1986 (1 December 1986) to 2 March 1992.*

7. Conclusions

In this paper we have reviewed the main issues associated with constructing and testing a set of successful trading models. We have presented results for one particular model which has sustained almost one year of *ex ante* testing for seven different FX rates. Our models are running real-time in continuing *ex ante* testing and our customers are actively following their recommendations. The usage of high frequency data ensures that the dealing prices used by our automatic traders are as close as possible to genuine market prices.

A systematic search for good indicators has allowed us to develop general yet high quality trading signals. The strategy employed by the trading models does not rely just on indicators (as is often the case in technical analysis models) but also on a set of rules designed to produce recommendations close to "real life" trading and which are easy for a FX dealer to track. The real-time processing of large quantities of data allows the models to adapt smoothly to rapidly changing market conditions. Extensive historical testing with a risk sensitive measure of performance reduces the danger of overfitting the parameters and gives rise to substantial returns that remain consistent over time.

We have developed a full trading model environment including a new data-driven programming language "G". This language permits us to develop trading models in a way that allows us to concentrate on building the models themselves rather than contend with issues essentially unrelated to the task at hand, as might well be the case were we employing a more conventional approach. The complexity of decision processes within a trading model requires a flexible scheduling system which we have implemented through a set of rules and meta-rules. The combination of a stable historical testing environment with real-time running of the models ensures that the results presented in this paper are not simply due to a fit of past data. Our models succeed in capturing a part of real market behavior.

This raises some interesting questions concerning conventionally accepted economic theories such as the "efficient market hypothesis" or the notion that market participants follow rational expectation strategies. Many economists regard the FX market as a paragon of efficiency. However, our models contradict this established view if we take efficiency to mean that market prices always fully reflect available information (see [6]) and that no profit can be made in the FX market from trading models relying alone on past prices.

Without going into details, we ask ourselves how our results fit in with these generally accepted theories. We contend that viewing the market as a set of actors all applying the same investment strategy is too simplistic and that further factors must be taken into account. An important parameter is the time horizon which varies for different investors. Clearly intra-day traders do not share the same inherent time horizons as central banks or long term strategic investors. Such differences give rise to effects that the efficient market theories do not address. We believe these are exactly the effects we have partly succeeded in catching and which make our trading models profitable.

In our future work, we intend to identify more accurately the characteristics of the different types of investors through research on the fundamental dynamics of the generating processes of price changes. We also intend developing more diversified trading models that correspond to the risk profile of each of the significant market components.

Acknowledgements

We thank O&A's Software and Technical Operation groups for making the price data available to us and for providing considerable and valuable assistance during the course of our work.

References

[1] Dacorogna M.M., Müller U.A., Nagler R.J., Olsen R.B., Pictet O.V.: A geographical model for the daily and weekly seasonal volatility in the FX market. To be published in Journal of International Money and Finance, 1992.

[2] Nagler R.J., Ward J.R.: G Users' Manual, 3rd Edition. Internal document RJN.1989-07-17, Olsen & Associates, Seefeldstrasse 233, 8008 Zürich, Switzerland, July 17, 1989.

[3] Müller U.A., Dacorogna M.M., Olsen R.B., Pictet O.V., Schwarz M., Morgenegg C.: Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis. Journal of Banking and Finance, 14:1189, 12 1990.

[4] Granger C.W.J., Newbold P.: Forecasting economic time series. Academmic Press, London, 1977.

[5] Keeney R.L., Raiffa H.: Decision with multiple objectives: Preferences and value tradeoffs. John Wiley & Sons, New York, 1976.

[6] Fama E.F.: Efficient capital markets: A review of theory and empirical work. Journal of Finance, 25:383-417, 5 1970.