THE MARKOVIANIZATION OF A LOOKBACK AMERICAN PUT OPTION PATH

Loreta Berhami
Marco Cora
Alessandro D'Arpa
Francesco Trusiano

January 12, 2006

Contents

1 A Review of the Problem

A lookback option is an option whose payoff process is path-dependent. In particular we are considering an American option, i.e. an option that does not have a particular moment in time in which it has to be exercised. The lookback option gives to the holder the opportunity to exchange the underlying asset with the highest price that the asset has had from the moment in which the option was subscribed till the moment in which it is exercised. The following formula expresses the payoff at time t of a lookback American put option, which is the case we are considering:

$$X(t) = \max_{0 \leq s \leq t} S(s) - S(t) \tag{1}$$

The first part at the right-hand side of the equality expresses the running maximum of S which is the strike price of the put option. While S(t) is simply the price of the underlying asset at the moment in which the option is generating an actual payoff, that is the option is being exercised. A lookback option is not a plain option to deal with because of the presence of the running maximum. In our example we are supposing that the underlying asset is a stock whose price (S) varies according to the usual binomial model. It increases (up factor) or decreases (down factor) with a certain rate and with a certain historical probability which are constant in any period we consider. In the usual binomial model the tree that represents the possible variations of the stock price S recombines which simplifies the problem considerably. In the case of the lookback option, the tree of the stock price remains unchanged from the simple binomial model, but the problem arises when we have to consider the tree of the values of the option itself. In order to calculate possible payoffs of the lookback option we have to add the running maximum to each cell of the tree. This variable makes the new tree not recombining anymore and this is a serious problem to deal with. In reality, as we will see later on, the tree does recombine; it is only much more difficult to find which model it follows. In order to be able to construct the new tree we have to markovianize the problem. Discretizing time as much as needed, we introduce another state variable instead of the running maximum:

$$F(t) = \max_{0 \leq s \leq t} S(s) \tag{2}$$

where:

$$F(t) = \max_{0 \le s \le t} S(s) = \max\Big\{\max_{0 \le s \le t} S(t); S(t+1)\Big\} = \max\{F(t); S(t+1)\}. \qquad (3)$$

It can be easily shown that this process is a Markovian process. Thus constructing a tree that collects in each cell the current values of S and F will allow us to fully evaluate the lookback American put option. Also in order to avoid the exponential growth of the number of cells (S, F) at time t, we set the rate of the price falling equal to the inverse of the rate of the price increasing, i.e. $d = \frac{1}{u}$.

Our original project involved the computation of the number of cells of the tree that collects the current values of S and F at time t. Some of these cells recombine. Our purpose is to write a code that makes the calculator count the number of cells that do not recombine, thus eliminating all the cells that are similar to at least one in a certain period of time. In order to achieve our goal we have to construct this tree, find the couples of cells that are identical (because the value of the underlying stock and of the running maximum is the same) and subtract this number from the number of the total cells that exist at period t. The latter number is given by 2^t. It is obviously a number of cells that tends to be not manageable as soon as the number of discretized periods of time becomes bigger than 25. The fact that the tree does recombine enormously simplifies the problem. In fact we will see later on that the number of cells which does not recombine and therefore have to be stored in memory is much smaller than the total number of cells generated by a not recombining tree.

Moreover we have decided to extend the project and allow our code to compute also the value of the option at time t, the hedging strategy and therefore the consumption of the writer if the option is not exercised optimally. Formally speaking, let us call the value of the option at time t, $V(t)$. At each time the holder of the option has to decide if he wants to exercise the option or if he wants to wait until at least the next period. Obviously, he will decide to exercise if the thinks that the value of the option will decrease and he will decide to wait if the thinks that the value of the option will increase in the next period of time. Therefore the value of the option is equal to the exercise value at instant t or to the continuation value, which in turn depends on the expected actualized value of the option in the future. From these considerations one can easily think of expressing the value of the option with the following recursive formula:

$$V(t) = \max\big\{F(t) - S(t); E^Q[\widetilde{V}(t+1)|S(t)]\big\} \tag{4}$$

At time T the value of the option is given by:

$$V(T) = \big[\max\{F(T-1); S(T-1)\cdot u\} - S(T-1)\cdot u\big]\cdot q + \big[F(T-1) - S(T-1)\cdot \tfrac{1}{q}\big]\cdot(1-q). \tag{5}$$

Once we have constructed the new tree with the current values of S and F, it is easy to compute according to the previous formulas the value of the option at each time t. The value of the American option has some particular properties which will be presented shortly after a few paragraphs. As for the hedging strategy, there are two ways of computing it.

One involves the use of the following formulas:

$$\vartheta_0 = \widetilde{V}(t) - \vartheta_1(t)\cdot\widetilde{S}(t), \tag{6}$$

$$\vartheta_1 = \frac{Cov_t^Q\big[\widetilde{V}(t+1), \Delta\widetilde{S}(t)|S(t)\big]}{Var_t^Q\big[\Delta\widetilde{S}(t)|S(t)\big]} \tag{7}$$

where ϑ_1 represents the number of units of the stock S to detain in a portfolio in order to hedge the lookback American put option and ϑ_0 is the number of units of the riskless bond. The tilda on the variables indicates that the values of these variables have been actualized. This method can be replaced by a much simpler to compute and more intuitive method which involves the solution of a system of two equations in which the unknown values are the two thetas (i.e. the hedging strategy).

This system basically expresses the deep meaning of the hedging strategy: given two types of assets in the market, one risky and one riskless, how can one construct a portfolio so that at each period of time he can reproduce exactly the same payoffs of the lookback American put option?

For example for the last period of the option the following system can be constructed:

$$\begin{bmatrix}(1+r)^T & S(T-1)\cdot u\\ (1+r)^T & S(T-1)\cdot\frac{1}{u}\end{bmatrix}\times\begin{bmatrix}\vartheta_0(T-1)\\ \vartheta_1(T-1)\end{bmatrix} = \begin{bmatrix}\max\{F(T-1); S(T-1)\cdot u\} - S(T-1)\\ F(T-1) - S(T-1)\cdot\frac{1}{u}\end{bmatrix} \tag{8}$$

Solving the above system is fairly easy.

The solution is the following:

$$\vartheta_0(T-1) = \frac{u}{u^2-1} \cdot \frac{F(T-1)\cdot u - \max\{F(T-1); S(T-1)\cdot u\}\cdot \frac{1}{u}}{(1+r)^T} \tag{9}$$

$$\vartheta_1(T-1) = \frac{u \cdot \left[\max\{F(T-1); S(T-1)\cdot u\} - S(T-1)\cdot u - F(T-1) + S(T-1)\cdot \frac{1}{u}\right]}{(u^2-1)\cdot S(T-1)} \tag{10}$$

If instead of solving the system we apply the (7) formula we obtain the following result:

$$\begin{aligned}\vartheta_1(T-1) =& \frac{\frac{u^2(1+r)-u(1+r)^2-u+(1+r)}{u^2-1}}{[(u-(1+r))^2\cdot q + (\frac{1}{q}-(1+r))^2\cdot(1-q)]\cdot S(T-1)}\cdot \\ &\cdot \frac{\left[\max\{F(T-1); S(T-1)\cdot u\} - S(T-1)\cdot u - F(T-1) + S(T-1)\cdot \frac{1}{q}\right]}{[(u-(1+r))^2\cdot q + (\frac{1}{q}-(1+r))^2\cdot(1-q)]\cdot S(T-1)}\end{aligned} \tag{11}$$

If we substitute the risk neutral probabilities q and $(1-q)$ with $q = \frac{u(1+r)-1}{u^2-1}$ and $1-q = \frac{u(u-(1+r))}{u^2-1}$ we obtain the same formula for $\vartheta_1(T-1)$that we obtained with the system. The same path can be followed in order to prove that the formula for $\vartheta_1(t)$ and $\vartheta_0(t)$obtained using (7) and (6) are identical from the ones obtained using the system (8) (See Appendix 4).

$$\begin{aligned}\vartheta_0(t) =& \frac{u}{u^2-1}\cdot\frac{\max\left\{F(t+1)-S(t)\cdot\frac{1}{u}; E^Q[V(t+2)]\right\}\cdot u}{(1+r)^{t+1}} + \\ &-\frac{u}{u^2-1}\cdot\frac{\max\left\{F(t+1)-S(t)\cdot u; E^Q[V(t+2)]\right\}\cdot\frac{1}{u}}{(1+r)^{t+1}}\end{aligned} \tag{12a}$$

$$\begin{aligned}\vartheta_1(t) =& \frac{u}{u^2-1}\cdot\frac{\max\left\{F(t+1)-S(t)\cdot u; E^Q[V(t+2)]\right\}}{S(t)} + \\ &-\frac{u}{u^2-1}\cdot\frac{\max\left\{F(t+1)-S(t)\cdot\frac{1}{u}; E^Q[V(t+2)]\right\}}{S(t)}\end{aligned} \tag{12b}$$

The hedging strategy as we have described it until now is not complete. In fact any American Option has a particular property. The actualized discounted value $\widetilde{V}$, also called Snell envelope of the discounted payoff process, is a super-martingale under Q[1], that is $\widetilde{V}(t_1) \leq E^Q\left[\widetilde{V}(t_2)|P_{t_1}\right]$ for any $t_1 \leq t_2$. This implies that the actualized discounted value of the American option is characterized by an early-exercise premium which decreases as time to maturity decreases. This, intuitively

---

[1] the risk neutral measure or equivalent martingale measure

speaking, is given by the fact that the more the holder waits the smaller is the time range within which he can maximize the value of its option. Moreover $\widetilde{V}$ is the smallest super-martingale greater or equal to the discounted payoff $\widetilde{X}$. These two features are particularly important from the point of view of the writer. This is because the discounted value of the American option $\widetilde{V}$ decreases in mean under Q and if the writer wants to hedge this option with a self-financing strategy (whose discounted value is a martingale under Q, i.e. has a constant mean), he will have to withdraw money from the self-financing hedging strategy. In order to find the amount of money that the writer has to withdraw from the self-financing strategy, first let us define the consumption process $C(t)$ as the cumulative consumption in the interval $[0;t)$. Also define $\Delta\widetilde{C}(t) = \widetilde{C}(t+1) - \widetilde{C}(t)$ as the discounted consumption in $[t;t+1)$ which is measurable with respect to $\mathcal{P}_t$ as the writer decides at the beginning of the period how much to consume in the period. The optimal consumption strategy paired with the self-financing hedging strategy is given by $\Delta\widetilde{C}(t) = -E^Q[\Delta\widetilde{V}(t)|P_t] \geq 0$.

The optimal exercise police, τ^* is defined as follows:

$$\tau^*(\omega) = \min\{t : \widetilde{V}(t)(\omega) = \widetilde{X}(t)(\omega)\}, \tag{13}$$

where $\widetilde{X}(t)$ is the discounted underlying payoff of the American option. If the holder exercises optimally in $[t;t+1)$ then the discounted value of the optimally exercised option,$\big\{\widetilde{V}(\min\{\tau^*(\omega);t\}(\omega))\big\}_{t=0}^{T}$, is a Q-martingale and thus no consumption is possible for the writer. On the contrary if the holder does not exercise optimally, the writer can consume a positive amount of wealth that corresponds to the early exercise premium that is lost by the holder in the period $[t;t+1)$. In our code we have calculated the consumption possible for any period of time and we have generated a vector of values for any possible path of variations followed by the underlying stock price S.

2 Close formula

The succession of the useful nodes is:

$$\{1, 2, 4, 6, 9, 12, 16, 20, 25, 30, \ldots\}$$

First we have to notice that a *pre-requisite* for the couple *stock and maximum* to recombine is that *at least* the stock recombines. For the commutative property of multiplication, it is obvious that the value of the stock after an increase and a decrease is equal to the value of the stock after a decrease and an increase. If moreover, we impose that:

$$d = \frac{1}{u}$$

then such values are not only equal among them, but they are equal to the value of the stock before any change took place. In general denoting with **u** an upward movement and **d** a downward movement, that if the number of **u**s is equal to the number of **d**s, the value of the stock will remain unchanged and the order of these movement will not count. For instance, the following path:

$$\{u, u, d, d, d, u, d\}$$

will be equal to:

$$\{u, d, d, u, d, u, d\} \quad \text{and} \quad \{u, d, u, u, d, d, d\}$$

Another interesting feature is that the number of recombining cells (for stock value) at each node follows a ***Tartaglia Triangle.*** Adding the constraint that the maximum has to recombine too, another regularity comes out: the number of cells which recombine follows this "strange" ***Tartaglia Triangle:***

$$\begin{array}{ccccccccc} & & & & 1 & & & & \\ & & & 1 & & 1 & & & \\ & & 1 & & 2 & & 1 & & \\ & 1 & & 2 & & 2 & & 1 & \\ 1 & & 2 & & 3 & & 2 & & 1 \end{array}$$

The regularity here is obvious: from the lefthand to the righthand the numbers increase up to a maximum and then start to decrease. There is a difference between even and uneven number of steps: in the first case the maximum number is repeated before the succession starts to decrease and in the latter it is not.

By looking at the tree we can notice that the succession is also the following series:

$$1 + 1 + 2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + \ldots$$

A recursive formula for the n^{th} element is:

$$\begin{cases} \psi(1) \leftarrow 1 \\ \psi(n) \leftarrow \displaystyle\sum_{w \in (1 \ldots n)} w - \psi(n-1) = \frac{n \cdot (n+1)}{2} - \psi(n-1) \end{cases}$$

To solve this *partial difference equation* we could use the tools mathematics provides us with or go through the following procedure. We can first analyze the succession of first and second differences:

$$\begin{aligned} \{\psi(n)\}_{n=1,2,\ldots} &= \{1, 2, 4, 9, 12, 16, 20, 25, 30, \ldots\} \\ \{\Delta\psi(n)\}_{n=1,2,\ldots} &= \{1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, \ldots\} \\ \{\Delta\Delta\psi(n)\}_{n=1,2,\ldots} &= \{1, 0, 1, 0, 1, 0, 1, 0, \ldots\} \end{aligned}$$

The last succession is an harmonic around $\frac{1}{2}$ and thus can be written as follows:

$$\{\Delta\Delta\psi(n)\}_{n=1,2,\ldots} = \frac{1}{2} - (-1)^n \cdot \frac{1}{2}$$

Proceeding backwards, the series of this succession can be created to obtain the first difference succession. The same procedure can be followed to obtain the original succession.

$$\begin{aligned} \{\Delta\psi(k)\}_{k=1,2,\ldots} &= \sum_{i=1}^{k} \{\Delta\Delta\psi(i)\}_{i=1,2,\ldots} = \frac{k}{2} + \frac{1}{2}\left[\frac{1}{2} \cdot (1 - (-1)^k)\right] \\ \{\psi(n)\}_{n=1,2,\ldots} &= \sum_{i=1}^{n} \{\Delta\psi(i)\}_{i=1,2,\ldots} = \frac{1}{2}\sum_{i=1}^{n} k + \frac{1}{2} \cdot \frac{n}{2} + \frac{1}{8} \cdot (1 - (-1)^n) \\ &= \frac{2n \cdot (n+2) + 1 - (-1)^n}{8} \end{aligned}$$

A final interesting feature is that:

$$\lim_{n \to \infty} \frac{\psi(n)}{\sum_{i=1}^{n} i} \to \frac{1}{2}$$

This is interesting because it ensures the succession always grows at a speed which, at maximum, is half of the arithmetic succession.

3 Comments to relevant features of the code

The code contains many features linked with the programming tool we decided to use which we will skip in this paper. The most interesting issues are the dynamic allocation of the memory, the vectorization of the matrix of the stock value and the maximum, and the creation of a more efficient function that finds the maximum between two values. Hereby there is a comment of the function which calculates the number of useful cells:

```
void alberi::conta(){
  int j,k,i,conta=0;
  double *Z = new (nothrow)double[n+1];
  for (i = 2; i < n+1; i++){
   for (j = pow(2,i-1); j <= pow(2,i)-1; j++){
     for (k = j+1; k <= pow(2,i)-1; k++){
     if((absale(S[j]-S[k]))<FLT_EPSILON &&(absale(f[j]-f[k]))< FLT_EPSILON){
        conta++;
        break;
     }
     }
   }
  Z[i]=pow(2,i-1)-conta;
  conta=0;
  printf(" %d - %f\n",i,Z[i]);
  }
}
```

The counter *conta* increases by 1 every time the value of the stock and the value of the maximum are equal. The functions $absale(S[j] - S[k]) < FLT\_EPSILON$ and $absale(f[j] - f[k]) < FLT\_EPSILON$ are needed because often the propagation of errors create a difference which is greater than the machine precision. By taking the absolute value of the difference only if it is smaller than the machine's precision we can avoid the problem of two *float* numbers comparison. Finally we subtract to the maximum number of cells per each time t the number of cells counted, thus obtaining the useful cells.

4 VBA translation

We finally translated the code into VBA (Visual Basic for Applications) to analyze the computational gains of using a tool which is nearer to the machine. As discussed in laboratory the code in VBA is much easier to read because it uses pre-defined formulas and it is shorter to write since many technical problems solved manually in C++ are already implemented into VBA.

These are the results:

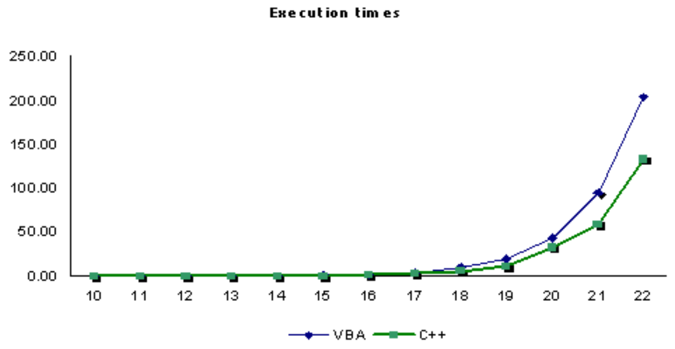

Figure 1: *Execution times*

The easiness of use of VBA is paid by a longer computation time needed as summarized by the following table:

n	10	11	12	13	14	15	16	17	18	19	20	21	22
VBA	0.02	0.05	0.09	0.18	0.39	0.88	1.92	4.19	9.15	19.94	43.47	94.39	204.43
C++	0.01	0.01	0.03	0.06	0.15	0.39	1.01	2.41	5.42	11.39	31.59	58.07	132.03
Ratio	0.480	0.236	0.301	0.352	0.389	0.445	0.524	0.575	0.592	0.571	0.727	0.615	0.646

Table 1: VBA and C++ performances in seconds

As we can see the loss in performance is the following: C++ usually takes between 40% and 60% the time it takes VBA to analyze the same tree.

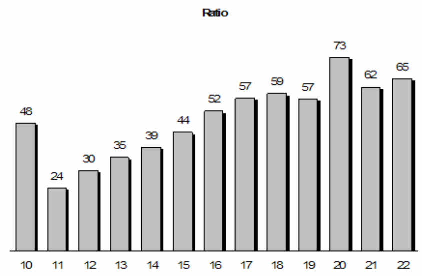

Figure 2: *Ratio of execution times*

Appendix A

$$\vartheta_1 = \frac{Cov_t^Q\big[\widetilde{V}(t+1), \Delta\widetilde{S}(t)|S(t)\big]}{Var_t^Q\big[\Delta\widetilde{S}(t)|S(t)\big]} \tag{14}$$

where:

$$\widetilde{V}(t+1) = \begin{cases} \max\big\{\widetilde{F}(t+1) - \widetilde{S}(t)\cdot u; E^Q[\widetilde{V}(t+2)]\big\} & \text{with prob } q, \\ \max\big\{\widetilde{F}(t+1) - \widetilde{S}(t)/u; E^Q[\widetilde{V}(t+2)]\big\} & \text{with prob } 1-q. \end{cases} \tag{15}$$

we consider first the numerator of the formula (14)

$$Cov_t^Q\big[\widetilde{V}(t+1),\Delta\widetilde{S}(t)\big] = \frac{\overbrace{\max\big\{F(t+1) - S(t)\cdot u; E^Q[V(t+2)]\big\}}^{max1}}{(1+r)^{t+1}} \cdot \left(\frac{S(t)\cdot u}{(1+r)^{t+1}} - \frac{S(t)}{(1+r)^t}\right)\cdot q \tag{16}$$

$$\cdot\frac{\overbrace{\max\big\{F(t+1) - S(t)\cdot \frac{1}{u}; E^Q[V(t+2)]\big\}}^{max2}}{(1+r)^{t+1}} \cdot \left(\frac{S(t)\cdot \frac{1}{u}}{(1+r)^{t+1}} - \frac{S(t)}{(1+r)^t}\right)\cdot (1-q) = \tag{17}$$

$$= \frac{1}{(1+r)^{2(t+1)}}\cdot\Big\{max1\cdot(S(t)\cdot u - S(t)(1+r))\cdot q + max2\cdot\big(\tfrac{S(t)}{u} - S(t)(1+r)\big)\cdot(1-q)\Big\} = \tag{18}$$

$$= \frac{S(t)}{(1+r)^{2(t+1)}}\cdot\Big\{max1\cdot(u-(1+r))\cdot q + max2\cdot\big(\tfrac{1}{u} - (1+r)\big)\cdot(1-q)\Big\}. \tag{19}$$

$$Var_t^Q[\Delta\widetilde{S}(t)] = \left(\frac{S(t)\cdot u}{(1+r)^{t+1}} - \frac{S(t)}{(1+r)^t}\right)^2\cdot q + \left(\frac{S(t)\cdot\frac{1}{u}}{(1+r)^{t+1}} - \frac{S(t)}{(1+r)^t}\right)^2\cdot(1-q) = \tag{20}$$

$$= \frac{S^2(t)}{(1+r)^{2(t+1)}}\cdot\Big\{(u-(1+r))^2\cdot q + \big(\tfrac{1}{u} - (1+r)\big)^2\cdot(1-q)\Big\}. \tag{21}$$

Given that

$$q = \frac{u(1+r)-1}{u^2-1} \qquad \text{and} \qquad 1-q = \frac{u(u-(1+r))}{u^2-1} \tag{22}$$

the coefficients of $max1$ and $max2$ can be rewritten as follows:

$$(u-(1+r))\cdot\frac{u(1+r)-1}{u^2-1} = \frac{u^2(1+r) - u(1+r)^2 - u + (1+r)}{u^2-1} \tag{23}$$

$$\left(\frac{1}{u} - (1+r)\right)\left(\frac{u(u-(1+r))}{u^2-1}\right) = -\frac{u^2(1+r) - u(1+r)^2 - u + (1+r)}{u^2-1} \tag{24}$$

Going back to the original formula, we have that:

$$\frac{Cov_t^Q\left[\widetilde{V}(t+1), \Delta\widetilde{S}(t)\right]}{Var_t^Q[\Delta\widetilde{S}(t)]} = \frac{1}{S(t)} \cdot \frac{max1(u-1-r)\cdot q + max2(\frac{1}{u}-1-r)(1-q)}{(u-1-r)^2\cdot q + (\frac{1}{u}-1-r)^2(1-q)} = \tag{25}$$

$$= \frac{1}{S(t)} \cdot \frac{-\frac{u^2(1+r)-u(1+r)^2-u+(1+r)}{u^2-1}\cdot(max1-max2)}{\frac{(u^2+1)(1+r)-u(1+r)^2-u}{u}} \tag{26}$$

The denomenator of the formula (25) can be simplified as follows:

$$(u-1-r)^2 \cdot \frac{u(1+r)-1}{u^2-1} + \left(\frac{1}{u}-1-r\right)\cdot\left(\frac{u(u-(1+r))}{u^2-1}\right) = \tag{27}$$

$$= \frac{(u^2+1)(1+r)-u(1+r)^2-u}{u} = \frac{u^2(1+r)-u(1+r)^2-u+(1+r)}{u} \tag{28}$$

Substituting (28) into (25), we obtain:

$$\vartheta_1(t) = \frac{Cov_t^Q\left[\widetilde{V}(t+1), \Delta\widetilde{S}(t)\right]}{Var_t^Q[\Delta\widetilde{S}(t)]} = \frac{u}{u^2-1}\cdot\frac{max1-max2}{S(t)} \tag{29}$$

If we use the following linear system, we obtain the same final solution for $\vartheta_1(t)$:

$$\begin{bmatrix} \overbrace{(1+r)^{t+1}}^{a} & \overbrace{S(t)\cdot u}^{b} \\ \underbrace{(1+r)^{t+1}}_{a} & \underbrace{S(t)/u}_{c} \end{bmatrix} \times \begin{bmatrix} \overbrace{\vartheta_0(t)}^{x} \\ \underbrace{\vartheta_1(t)}_{y} \end{bmatrix} = \begin{bmatrix} \overbrace{max1}^{d} \\ \underbrace{max2}_{e} \end{bmatrix} \tag{30}$$

$$y = \frac{e-d}{c-b} \qquad x = \frac{d\cdot c - b\cdot e}{(c-b)\cdot} \tag{31}$$

$$\vartheta_1(t) = \frac{max2-max1}{\frac{S(t)}{u} - S(t)\cdot u} = \frac{1}{S(t)}\cdot\frac{max2-max1}{\frac{1}{u}-u} \tag{32}$$

$$\vartheta_1(t) = \frac{u}{u^2-1}\cdot\frac{max1-max2}{S(t)} \tag{33}$$