DELPHI LABS

Presents.

Dynamic Interest Rate Model — Based on Control Theory

DeFi Credit Protocols: Dynamic Interest Rates Using Control Theory

Ayana T. Aspembitova

May 2021

1 Introduction

Credit Protocols such as Aave and Compound have been developing as one of the largest verticals within Decentralized Finance. These automated, non-custodial credit facilities match users who want to borrow funds with users looking to earn yield on idle assets by lending them out. Within credit protocols, interest rate models play a crucial role - they balance supply and demand, ensuring that depositors' funds are effectively utilized and that the protocol remains liquid. In traditional finance interest rates are primarily set by central banks and calculated based on the maturity of the loan position. Although some protocols are quite similar to the traditional finance borrowing and provide fixed rate loans with the fixed maturity, the most prominent protocols work in fundamentally different way - there is no maturity date in ledger loans and the ability to borrow assets depends on the liquidity in the pool.

2 Interest rate models in lending protocols

Currently interest rate models used in Credit Protocols are some linear or non-linear function of the available funds in a given market. Specifically, utilization of deposited funds is the main factor used to estimate the interest rates. Utilization U for a given market m can be defined as follows:

$$U_m = \frac{L}{A} \tag{1}$$

where L is total loans and A is gross deposits.

Interest rate models need to balance the twin goals of liquidity and capital efficiency. Firstly, U_m must not exceed 100% as this represents full utilization with the protocol being solvent but illiquid, as users are unable to withdraw funds. Secondly, the protocol must ensure U_m is not too low as this leads to capital inefficiency. Later we review all existing models used currently in credit protocols and how they solve the optimal utilization problem.

2.1 Linear Model with a kinked rate

This is the most widely used type of model - interest rates are set as a linear function of utilization and they change sharply once a certain threshold of utilization is reached (the target utilization). Mathematically, they can be represented as follows:

$$i_b = \begin{cases} \alpha + \beta U, & \text{if } U \leq U_o \\ \alpha + \beta U_o + \gamma(U - U_o), & \text{if } U > U_o \end{cases} \tag{2}$$

where U is the utilization ratio, U_o is the optimal utilization, α is the base rate and β is the per-block multiplier.

This type of model provides incentives for borrowers to repay their debt once the optimal utilization is reached and encourages depositors to add funds to the pool by sharply increasing the interest rate. A linear model with a kinked rate is currently used by Aave and Compound.

Always changing interest rates could be a disadvantage for borrowers and depositors who seek for stability and foreseeable payments. Aave protocol offers the option for users to choose a stable interest rate. Stability is achieved by computing the protocol-wide market rate m_r as the mean of the total borrowed funds by the borrow rate for given protocol p:

$$m_r = \frac{\sum_{p=1}^{n} i_{b,m,p} \cdot B_{m,p}}{\sum_{p=1}^{n} B_{m,p}} \tag{3}$$

where $B_{m,p}$ is the total amount of borrowed funds on market m on lending protocol p.

Using the market rate as a base, the borrow interest rate is calculated as follows:

$$i_{b,m,s} = \begin{cases} m_r + \frac{U}{U_o} \cdot R_{slope1}, & \text{if } U < U_o \\ m_r + R_{slope1} + \frac{U-U_o}{1-U_o}, & \text{if } U \geq U_o \end{cases} \tag{4}$$

As we can see in Equation 4, the stable interest rate in Aave is not entirely stable - it changes sharply when utilization reaches the threshold in the same way as linear rates.

2.2 Non-linear Interest Rate Model

Interest rates can also be set as a nonlinear function of utilization and computed as:

$$i_b = (\alpha \cdot U) + (\beta \cdot U^n) + (\gamma \cdot U^m) \tag{5}$$

where n and m can be set depending on how aggressive interest rate needs to grow when utilization goes up. This model allows to increase the interest in a non-linearly increasing rate and is currently used in dYdX.

2.3 Areas for improvement in current interest rate models

Existing interest rate models have been able to provide a decent experience to users, achieving large TVL (Total Value Locked). Still, there is room for improving the effectiveness and safety in many ways. First, we will show the main problems that credit protocols have experienced so far, then discuss some possible risks and conclude by proposing a solution.
Since the interest rates at time t are calculated as a function of utilization at the time t, they can vary greatly every day. Figure 1 shows borrow rates' evolution in USDT market in Aave V2. Although users in credit protocols expect uncertainty to some extent, having less volatile interest rates could improve their experience and help to gain broader adoption.

Sometimes popular markets in credit protocols experience periods of full illiquidity when all funds are utilized. For example, Figure 2 shows that in USDT markets the utilization is frequently above the optimal 90%. Borrowers are not incentivised to repay their debt, which is mostly caused by some external factors - for example, there are opportunities in DeFi to earn higher returns on the assets they borrow from credit protocols. These external factors also can be a reason that depositors would want to withdraw their funds to get higher income outside of the credit protocol. This could be an undesirable scenario for the protocol's liquidity, since current interest rate models do not consider external factors and fixed interest rate model cannot adapt to the overall DeFi conditions. Moreover, the heavy-tailed distribution of depositor's funds found in Compound [4] and Aave markets (Figure 3) could make the periods of high utilization riskier, as 30-50% of protocol's liquidity provided only by a few depositors.

Another area for improvement is markets with low utilization. As some assets are regularly more popular to borrow, while others are mostly used as collateral, a pattern emerges where some markets are consistently underutilized. For example, the BAT market in Aave has utilization below 20% most of the time (Figure 2). Incentivising the higher demand in non-popular markets would improve the capital efficiency and overall protocol health.

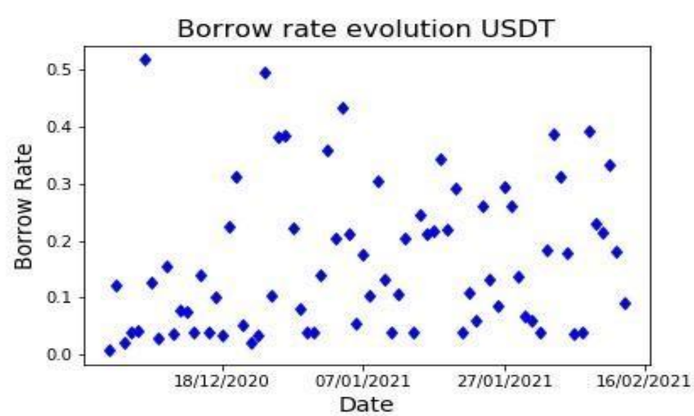

Figure 1: USDT market Borrow rates in Aave V2 protocol.

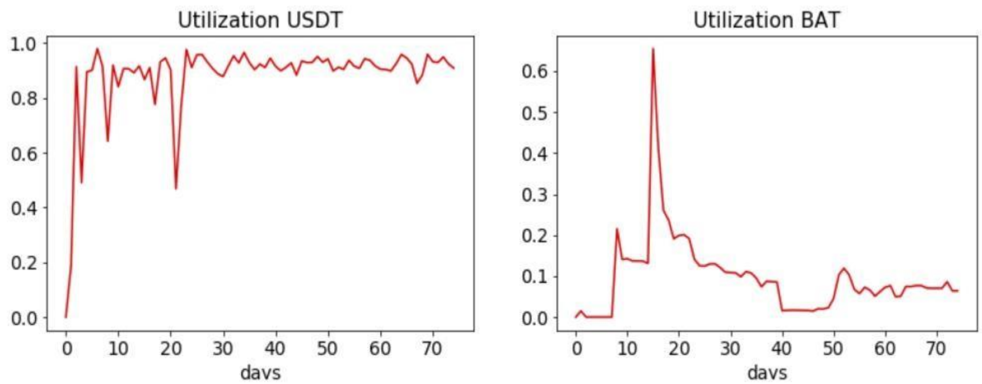

Figure 2: Utilization rates in USDT and BAT markets in Aave V2 protocol.

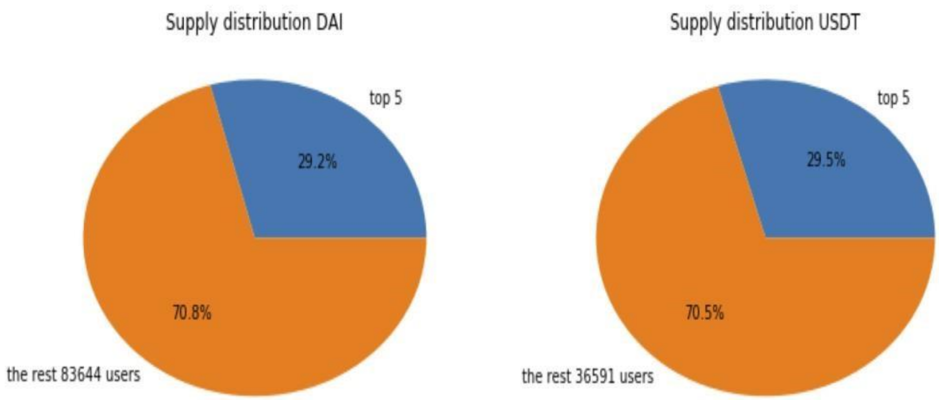

Figure 3: Distribution of supplied amounts among users in USDT and DAI pools.

All these areas can be improved by achieving a more responsive, stable utilization rate. In the next section we propose new type of interest rate model that would help to achieve this goal and eliminate some of the previously explored economic risks.

3 Dynamic Interest Rate Model based on the PID Controller

Constant (or close to constant) utilization is a highly desirable goal for lending protocols as it ensures the stability of the protocol by reducing illiquidity risk. Achieving this goal is not a trivial task, since the cryptocurrency market is quite dynamic and demand for assets can vary greatly.
We explore alternative models beyond finance and economics theories to achieve the goal of more stable utilization rates.

This idea is to look at the utilization as a parameter that can be manipulated by another parameter (for example, temperature in physical systems can be manipulated using heating and cooling units). Keeping a response parameter constant is a common problem in control theory and there is a well-known method to solve it - the PID controller (proportional-integral-derivative controller) and its variations. A PID controller continuously calculates the difference between the desired point (set point - optimal utilization) and actual value (process variable - actual utilization). Then it continuously corrects itself based on the proportional, derivative, and integral terms. Each term has its own purpose. The proportional term provides a corrective response based on the error between desired and actual value. The integral term accounts for the past errors and integrates over them in order to eliminate residual error. Finally, the derivative term estimates the future trend of the error based on its current rate of change. The overall control function looks as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt} \tag{6}$$

where $e(t)$ is the error between set point and actual value, K_p is the proportional parameter, K_i is the integral parameter and K_d is the derivative parameter.

3.1 Implementation and Parameter Tuning

We now propose a way to implement this method in Credit Protocols. We assume that interest rate is the main factor that drives the utilization set that as the *control parameter* in our model. With utilization being our *response parameter*. The overall algorithm to calculate the borrow interest rate can be seen in Figure 4.

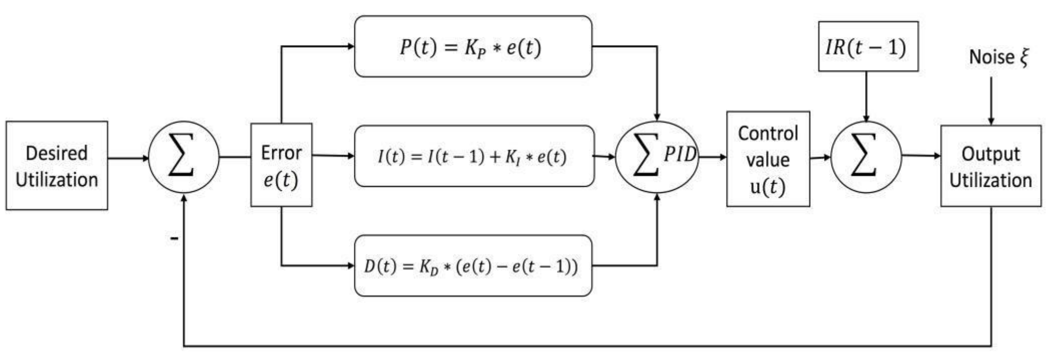

Figure 4: Algorithm to calculate the Borrow Interest Rate

First, the difference between the desired (optimal) utilization U_o and the actual utilization is calculated as $e(t) = U_o - U(t)$. Then we calculate three functions - proportional, integral, derivative and sum their outputs. The obtained value $u(t)$ is the control value that needs to be applied to the interest rate in order to stabilize the utilization.

To understand how markets will behave when using the dynamic model, we first empirically derive the relations between the change in interest rates and the market response from Aave V2 data. In Aave interest rate is calculated as shown in Equation 2 - at time t interest rate reflects the utilization *U(t)*. Change in utilization in the next time step *U(t)-U(t+1)* then reflects the response of market to the change in interest rate in previous time step *IR(t-1)-IR(t)*. We obtain two time series from empirical data - changes in interest rates and lagged changes in utilization. This allows us to understand how responsive is market - we checked on Aave USDT, BAT and SNX markets. The general relations between the change in interest rate and the market response can be expressed as in Equation 7:

$$U(t) = U(t-1) + \Delta IR \times \omega + \xi \tag{7}$$

where $\triangle IR$ is the change in interest rate.

The term ω defines how quick the utilization rate reacts to the change in interest rate. For popular markets we expect the decrease in interest rate to lead to a quick growth in utilization, while for unpopular markets the change would be smaller. ξ is a factor influencing the utilization rate, otherwise known as noise.

The next step is to obtain the optimal parameters for the PID functions (K_p, K_i, K_d). This is not a trivial task for any PID controller implementation, and there are many methods and theories developed for optimal parameter tuning. The overall effects of the parameters are as follows. The larger K_p and K_i the faster optimal utilization is reached, but also the larger the probability of overshooting the target. The larger K_d, on the other hand, the lower the probability of overshooting the target, acting as a balancing parameter in conjunction with the two previous ones.

We use the standard Ziegler-Nichols method to tune the parameters. We switch o_ the integral and derivative functions and keep increasing the parameter K_p until we achieve stable oscillations in utilization - ultimate gain K_u at the time T_u. Then we set the parameters according to Table 1.

Control type	K_p	K_i	K_d
PID	$0.6K_u$	$1.2K_u/T_u$	$0.105K_uT_u$
No overshoot	$0.2K_u$	$0.4K_uT_u$	$0.066K_uT_u$

Table 1: Optimal parameters for PID tuning

With the parameters defined we simulate the evolution of the utilization rate and interest rates. We start from a utilization of 60% and a borrow interest rate of 10%, while trying to achieve 90% target utilization. Then we calculate the control parameter as shown in Figure 4 and change the interest rate accordingly. The response utilization rate is simulated according to Equation 7. We set the noise to be small to assess the general behavior of the new model.

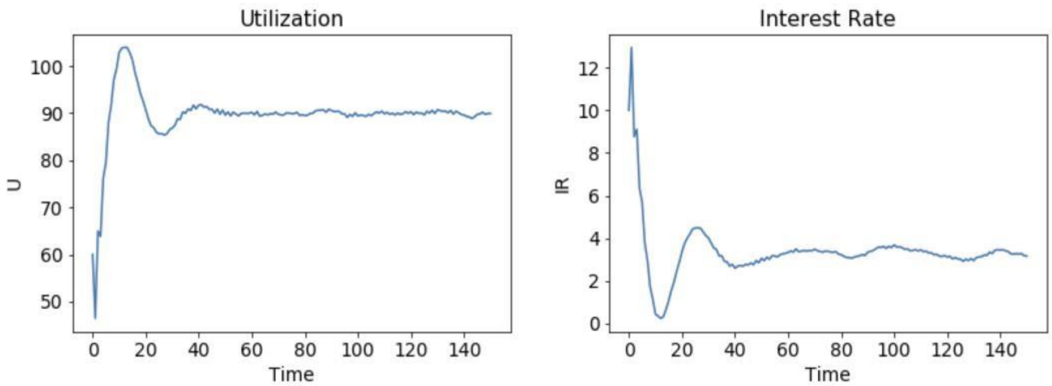

Figure 5: Utilization and Interest Rate evolution for markets with small response rate, K_u= 10, K_p= 6, K_i= 2, K_d= 12.75

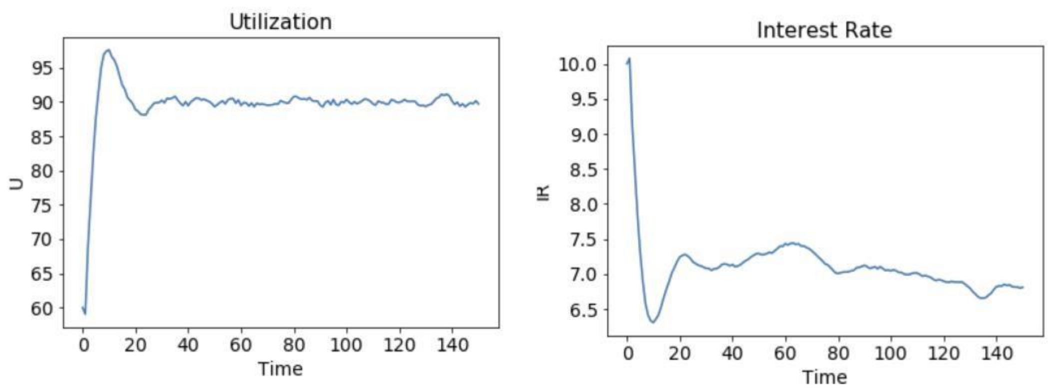

Figure 6: Utilization and Interest Rate evolution for markets with large response rate, K_u= 5, K_p= 3, K_i= 1, K_d= 2.25

Figures 6 shows the evolution of utilization and interest rates for markets with high (eg. stable- coins) and low (e.g. BAT) response rates. Within 10-time steps utilization increases, but the model overshoots the set point (90%) in both types of markets. The model then increases interest rates, driving utilization down. After one cycle of oscillations we achieve the desired utilization (under assumption there is little noise). As for the interest rate evolution, they fluctuate a lot while trying to achieve U_o, but once we obtain 90% of utilization, borrow rates are more stable than when using the linear or non-linear models. However, this is only the case if we are looking at a closed system without any alternative protocols or external factors and little noise. In reality we expect to have more volatile interest rates, as we try to stabilise the utilisation.

We have shown how the PID Controller can be implemented to achieve stable utilization in different markets. The simulation shown above was conducted under the strong assumptions that there is little noise and no strong external factors to demonstrate the logic of the model. This model can also be modified in order to adapt to various market conditions - in the next section we propose some possible improvements to this model to increase its effectiveness.

3.2 External Factors and Noise

In our previous simulations we assumed that there is only a small amount of noise and the change in utilization mostly depends on the change in interest rate. However, in real markets external factors may strongly influence utilization - for example, if there are some good farming opportunities on a certain asset, the demand will grow and the response rate would be small. Or the opposite situation, if the protocol provides additional incentives for depositors by offering extra rewards, then the supply will grow, taking utilization lower. To ensure that our model will behave properly under these conditions we will be taking the following actions:

1. Freeze the integral term
2. Set proportional parameter K_p higher.

The first step is necessary to avoid the compensation from the integral term for the increasing error by introducing another error in the opposite direction. This could lead to the large overshoot and by ignoring the input from the integral function we would eliminate the wrong estimation of control function. The second step would help us to bring the utilization to optimal faster - when there are strong external factors, users react less to the changes in interest rates. Therefore, by increasing them drastically we can increase the response rate and avoid the full illiquidity (or very low utilization, depending on the external factor).

Apart from external factors, noise can also affect the behavior of the model. Noise can be driven by certain random user behaviors and can lead to wrong estimations of the control function and difficulties stabilizing the utilization rate. We plan to remove the noisy effect of random users by 6 filtering the actual utilization U_t.. Instead of calculating the error as $e(t) = U_t - U_o$, we use the error between the filtered utilization and optimal $e(t) = U_t^f - U_o$. Depending on the nature of noise in various markets, we set the type of filter accordingly. Another strong factor that will constantly influence the model's performance is the external interest rates - there are several credit protocols on different blockchains and interest rates offered by them could influence the overall responsiveness of users and the volatility of the interest rates in a given protocol.

4 Implementation

We plan to implement the first iteration of our Dynamic Interest Rate model on Mars Protocol, a Credit Protocol built on the Terra blockchain that is currently being incubated by Delphi Labs and IDEO CoLab Ventures. Although our preliminary simulations show that it is possible to safely implement the dynamic model, we would take precautions and start using only the proportional (P) term. This would allow us to avoid large overshoots due to the error compensation by the integral term as well as possible high frequency noise caused by the derivative term.

In addition, we will set the condition that once the error e reaches a certain threshold (meaning that there is large influence of the external factors), we increase the parameter K_p sharply. At the start, the behavior of utilization and interest rates might be similar to the current Aave and Com- pound markets with high demand. Later, after we obtain the hard data about the noise, response rate and the utilization rate trend, we would iterate and improve the model by introducing the integral and derivative terms together with the filter, as described above

```
pub fn canonical_pid_implementation<S:Storage>(
    storage: &mut S,
    asset_reference: &[u8],
) -> StdResult<()> {
    let pid_parameters = load_pid_parameters(storage, asset_reference)?;
    let debt = load_debt(storage, asset_reference)?;
    let liquidity = load_liquidity(storage, asset_reference)?;

    let u: i32 = debt / liquidity * 10000;

    let error = pid_parameters.u_optimal;

    let p = pid_parameters.kp * error;
    let i = pid_parameters.previous_i + error * pid_parameters.ki;
    let d = (error - pid_parameters.error) * pid_parameters.kd;

    let previous_interest_rate = load_interest_rate(storage, asset_reference)?;

    let mut new_interest_rate =  previous_interest_rate + p + i + d;

    if new_interest_rate < 0 {
        new_interest_rate = 0;
    }

    pid_parameters.previous_i = i;
    pid_parameters.previous_error = error;

    save_pid_parameters(storage, asset_reference, &pid_parameters)
    save_interest_rate(storage, asset_reference, new_interest_rate)?;
    Ok(())
}
```

Figure 7: Canonical implementation of the PID controller model to adjust the interest rates. Code is written in Rust and can potentially be used by any CosmWasm credit protocol.

We will also be proposing this model for consideration to the Aave and Alpha Homora communities. Overall, we believe this model is applicable to any lending protocol and are happy to work with teams on experimenting with its implementation.

5 Conclusions

The interest rate pricing model lies at the heart of any credit protocol. Ideally, it needs to provide the best experience for both depositors and lenders ensuring the protocol's capital efficiency and solvency. All of these goals can be achieved when there is a stable utilization ratio across markets in the protocol. Given the volatile nature of crypto assets, the fast-evolving space of Decentralized Finance and the unpredictable behavioral patterns of users, achieving stable utilization becomes a difficult task. To solve this problem, we develop a new interest rate model that is based on the feedback loop between desired and current utilization. By calculating proportional, integral and derivative functions we can teach" the model to adapt to constantly changing market conditions. Importantly, implementation of this model is quite simple and can be achieved in only a few lines of code, which is important for on-chain use. It is also flexible and can be adjusted for different markets. By introducing simple modifications, we are able to model various external factors that are currently not being considered by interest rate models in other credit protocols. We are excited to receive feedback and comments from the DeFi community and work with any teams who are interested in exploring this idea further.

6 Acknowledgements

We would like to acknowledge Michael Bentley from Euler XYZ for his valuable feedback and help revising this report. The initial idea of using a PID Controller within the lending context came from his work in Euler XYZ.

Additionally, we would like to thank Will Price and Vasiliy Shapovalov for their helpful feedback on this piece.

References

[1] Aave Protocol Whitepaper. January 2020

[2] R. Leshner, G. Hayes Compound: The Money Market Protocol. February 2019

[3] A. Juliano dYdX: A Standard for Decentralized Margin Trading and Derivatives. September 2017

[4] L. Gudgeon, S.Werner, D. Perez, W.J. Knottenbelt DeFi Protocols for Loanable Funds: Interest
Rates, Liquidity and Market Efficiency. arxiv pre-print October 2020

[5] J.G Ziegler, N. B. Nichols Optimum settings for automatic controllers. Transactions of the ASME, 1942

[6] B. C. Dorje, P.H. Lane, B. K. Meister Theory of Cryptocurrency Interest Rates. arxiv, 2019

[7] F. Rodriguez PID Schematic Implementation Code.
https://gist.github.com/spikeonmars/46b2c824d621b564a3b272f9c9b160a3 May 2021

[8] M. Bentley Introducing Euler. https://medium.com/euler-xyz/introducing-euler-8f4422f13848 December 2020

DELPHI DIGITAL

@DELPHI_DIGITAL

WWW.DELPHIDIGITAL.IO

CONTACT@DELPHIDIGITAL.IO