

# A Dynamic Fee Adjustment Model for Lightning Channels

Created by: feelancer21@github

## Summary

This model suggests a fee adjustment mechanism for lightning channels based on their liquidity state. Like any other model, it requires parameterization to control the speed of fee adjustment. Currently, these parameters need to be manually configured, but automatic recalibration may be possible in the future with more research. It is advisable to use this model only if you fully understand its implications.

The fee adjustment model aims to achieve the following objectives: 1. Controlling fee rates on the sinks to prevent rapid depletion. 2. Managing fee rates on the sources to cap rebalancing costs, considering the differences in feerates (spreads) between sinks and sources. 3. Maximizing profits by optimizing margins.

It's worth noting that optimizing profits (point 3) was not the primary focus in the current stage, but it may be addressed through proper model parameterization in the future.

## Overview of the Model Design

The model adjusts fee rates to balance channel liquidity effectively, utilizing two additive components:

1. The first component exponentially increases the feerate on a peer level when the channel balance leans toward the remote side and decreases it when it leans toward the local side. This component affects spreads and influences rebalancing costs.
2. The second component adjusts all node feerates by the same absolute value. If the sinks are more depleted than filled, it increases the margins; otherwise, it decreases it. This component doesn't impact spreads but provides more time for rebalancing, preventing rapid depletion.

## PID Controllers

Both components of the model are based on the concept of PID (Proportional-Integral-Derivative) Controllers. PID Controllers utilize a measured process variable, compare it to a target value, and calculate an error function, denoted as  $e(t)$ . They use a linear function to adjust the control variable based on the error, its integral over time, and its derivative, aiming to minimize the error in subsequent iterations.

## Usage and Modifications of PID Controllers

First, let's discuss the measured variable and error function definitions for each component, followed by modifications to classical PID Controllers:

1. For the first component, the remote balance is compared to a target value, typically derived from the average liquidity ratio of all channels. The difference between the observed remote balance and the target is mapped to an error ( $e$ ) in the range  $[-0.5, 0.5]$  using linear interpolation. When the remote balance equals the target,  $e$  is set to 0.
2. For overall depletion, two weighted feerate averages are compared: a) Weighted with the local balance and b) weighted by capacity multiplied by the target. The error is calculated as the difference between b) and a). A positive error indicates that sinks are more depleted and requires an increase in feerates, while a negative error suggests the opposite.

In this controller, classical integrals and derivatives are not used. Instead, an exponential weighted moving average with a smoothing parameter ( $\alpha_i$ ) defines the implicit length of the error history. Additionally, an exponential decay with a parameter ( $\alpha_d$ ) is applied to the error delta as a derivative component. You can also specify a drift for adjustments that scale over time but are not influenced by the error.

## Implementation Details

Let  $T$  be the current time, and  $T_0$  represent the oldest observed historic timestamp.

We define  $r_P(t)$  as the feerate for a peer  $P$  at time  $t$ , which can be decomposed as follows:

$$r_P(t) = x_S(t) + x_M(t)$$

The components  $x_S(t)$  and  $x_M(t)$  correspond to the previously mentioned component one and component two, respectively.  $x_M(t)$  is the margin which modelled by a mean reverting controller and is not peer dependent.  $x_S(t)$  is the peer dependent spread and modelled with the PID approach.

### Modelling the margin with a mean reverting controller

The margin  $x_M(t)$  is controlled by the following differential equation.

$$dx_M(t) = \alpha \cdot (K_m - x_M(t))$$

$K_m$  is called the mean reversion level. If  $K_m$  equals  $x_M(t)$  then  $dx_M(t) = 0$  and no further adjustments of the margin are needed.  $\alpha > 0$  is called the mean

reversion speed and determines how quick  $x_M(t)$  reverts to the mean reversion level  $K_m$

The solution of the differential equation for  $t > T_{n-1}$  with a given initial value  $x_M(T_{n-1})$  is

$$x_M(t) = K_m \cdot (1 - \exp(\alpha(T_{n-1} - t))) + x_M(T_{n-1}) \cdot \exp(\alpha(T_{n-1} - t))$$

or equivalent

$$x_M(t) = x_M(T_{n-1}) + (K_m - x_M(T_{n-1})) \cdot (1 - \exp(\alpha(T_{n-1} - t)))$$

### Modelling the spread with a PID controller

$$dx_S(t) = (T(t) + P(t) + I(t) + D(t))dt$$

This equation leads to:

$$x_S(T_n) = x_S(T_{n-1}) + \int_{T_{n-1}}^{T_n} (T(t) + P(t) + I(t) + D(t))dt$$

Where: -  $\int_{T_{n-1}}^{T_n} T(t) dt$ :  $T$  represents the error-independent drift. - Similar definitions apply to the parts  $P$ ,  $I$ , and  $D$ , which are the proportional, integral, and derivative parts of the controller.

Now, let's delve into the different parts. We will assume that the error is a constant value  $y$  for  $t \in ]T_{n-1}, T_n]$ .

**Drift and Proportional Part** These parts are relatively straightforward:

$$T = K_t \int_{T_{n-1}}^{T_n} dt = K_t(T_n - T_{n-1})$$

And

$$P = K_p \int_{T_{n-1}}^{T_n} e(t)dt = K_p(T_n - T_{n-1}) \cdot y$$

**Integral Part** In this section, we will discuss the calculation of an exponential weighted moving average (EWMA) in a continuous world for an integrable function  $e(t)$ . This allows us to derive a recursive formula in the discrete world, even when measuring points of the function are not necessarily equidistant.

Let  $\alpha \geq 0$  be the smoothing parameter. We define the exponential continuous moving average as follows:

$$\text{EWMA}_\alpha(T_n, T_{n-1}) = \int_{T_{n-1}}^{T_n} \alpha \exp(\alpha(\tau - T_n)) e(\tau) d\tau + \exp(\alpha(T_{n-1} - T_n)) \cdot \text{EWMA}_\alpha(T_{n-1}, T_{n-2})$$

By complete induction, it can be shown that for all  $k = 0, \dots, n-1$ :

$$\text{EWMA}_\alpha(T_n, T_{n-1}) = \text{EWMA}_\alpha(T_n, T_k)$$

This allows us to set:

$$\text{EWMA}_\alpha(t) = \text{EWMA}_\alpha(t, T_0)$$

Using the recursive definition, we can express it as:

$$E_\alpha(t) = \int_{T_{n-1}}^t \alpha \exp(\alpha(\tau - t)) e(\tau) d\tau + \exp(\alpha(T_{n-1} - t)) \cdot E_\alpha(T_{n-1})$$

Assuming a constant value  $e(\tau) = y$  since  $T_{n-1}$ , we obtain:

$$E_\alpha(t) = (1 - \exp(\alpha(T_{n-1} - t))) \cdot y + \exp(\alpha(T_{n-1} - t)) \cdot E_\alpha(T_{n-1})$$

We set the integral part of the controller as:

$$I = K_i \int_{T_{n-1}}^{T_n} E_\alpha(t) dt = K_i \left( T_n - T_{n-1} + \frac{1}{\alpha} (1 - \exp(\alpha(T_{n-1} - T_n))) \cdot (E_\alpha(T_{n-1}) - y) \right)$$

This allows us to update the controller recursively with only the knowledge of the value of  $E_\alpha(T_{n-1})$ .

**Derivative Part** For the derivative part, we use an exponential decay. Consider an error change  $e(T_k) - e(T_{k-1})$ , and for any  $n \geq k$ , let:

$$\begin{aligned} D_k &= (e(T_k) - e(T_{k-1})) \int_{T_{n-1}}^{T_n} \alpha \exp(\alpha(T_{k-1} - \tau)) d\tau \\ &= (e(T_k) - e(T_{k-1})) (\exp(\alpha(T_{k-1} - T_{n-1})) - \exp(\alpha(T_{k-1} - T_n))) \\ &= (e(T_k) - e(T_{k-1})) \exp(\alpha(T_{k-1} - T_{n-1})) (1 - \exp(\alpha(T_{n-1} - T_n))) \end{aligned}$$

The derivative part of the controller is set as:

$$D = K_d \sum_{k \leq n} D_k$$

In the end,

$$\int_{T_{k-1}}^t \alpha \exp(\alpha(T_{k-1} - \tau)) d\tau = 1 - \exp(\alpha(T_{k-1} - t))$$

converges to 1, and thus, the entire error change decays over time.