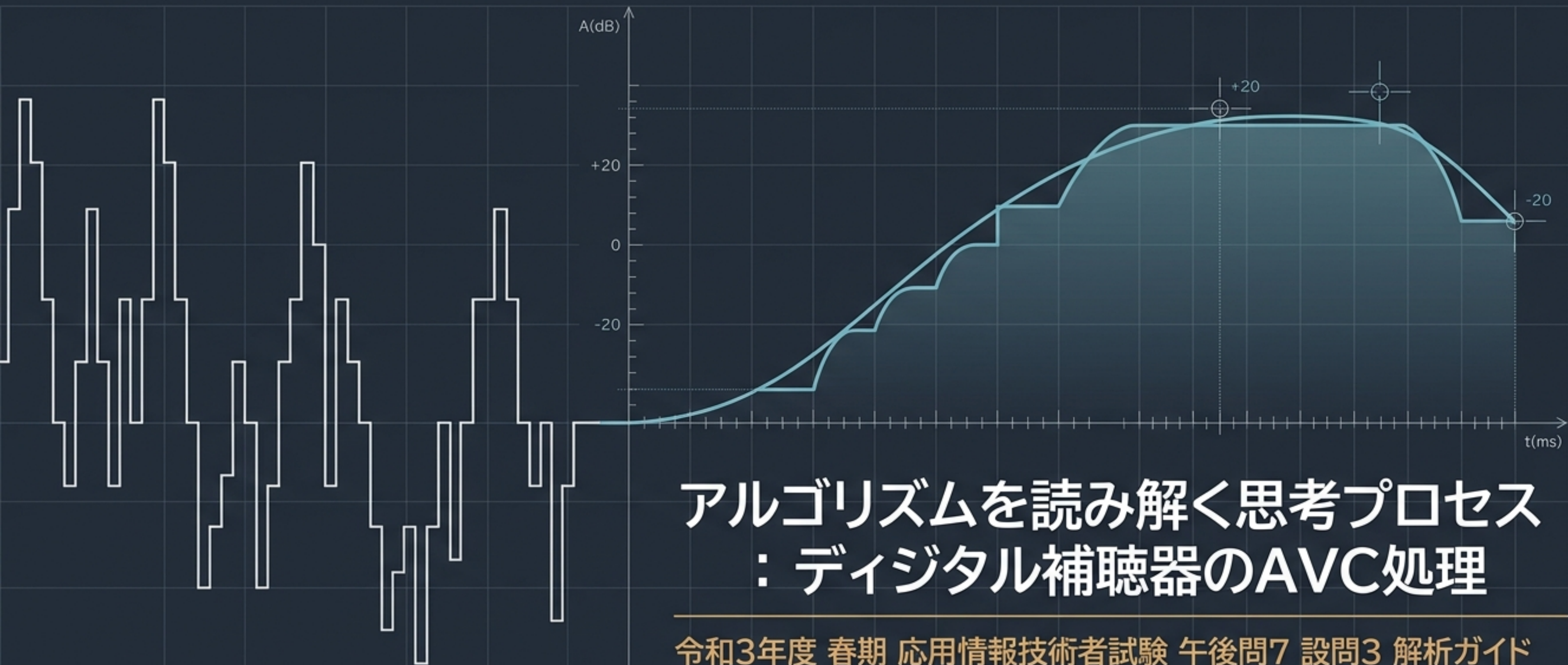




アルゴリズムを読み解く思考プロセス ： デジタル補聴器のAVC処理

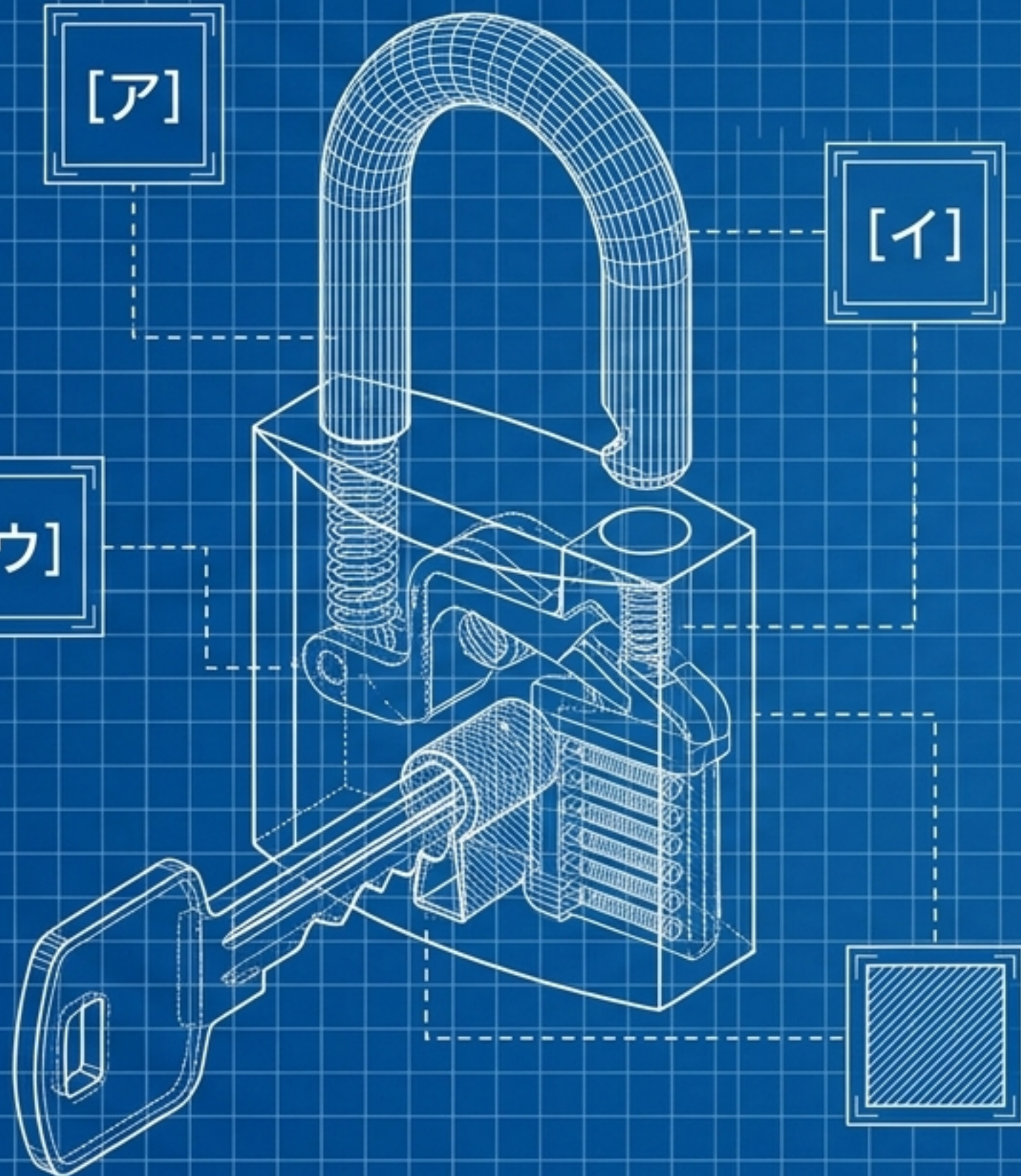
令和3年度 春期 応用情報技術者試験 午後問7 設問3 解析ガイド

※本スライドに解答は直接記載していません。
自力で正解に辿り着くための「思考の足場」を提供するガイドです。

[ア]

[イ]

[ウ]



答えではなく、「解き方」の型 を手に入れる

[ミッション]

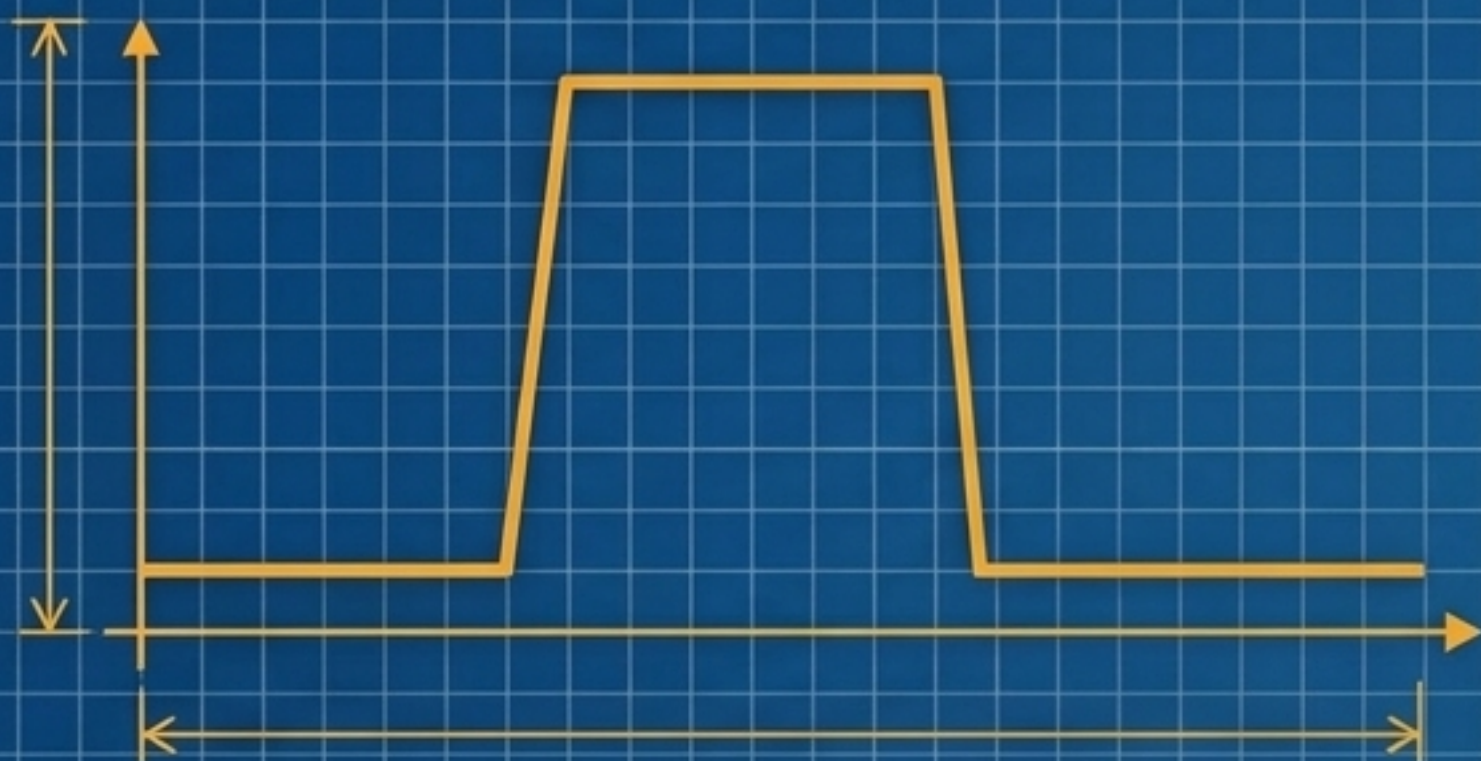
AVC（自動音量調節）のフローチャートの空欄を補完し、特定の条件で発動する例外処理の意図を自らの言葉で言語化すること。

[アプローチ]

正解の文字列を暗記しても応用力は身につきません。本資料ではフローチャートを解剖し、変数の物理的な振る舞いを可視化することで、エンジニアの設計意図を「完全に理解」する状態へと導きます。

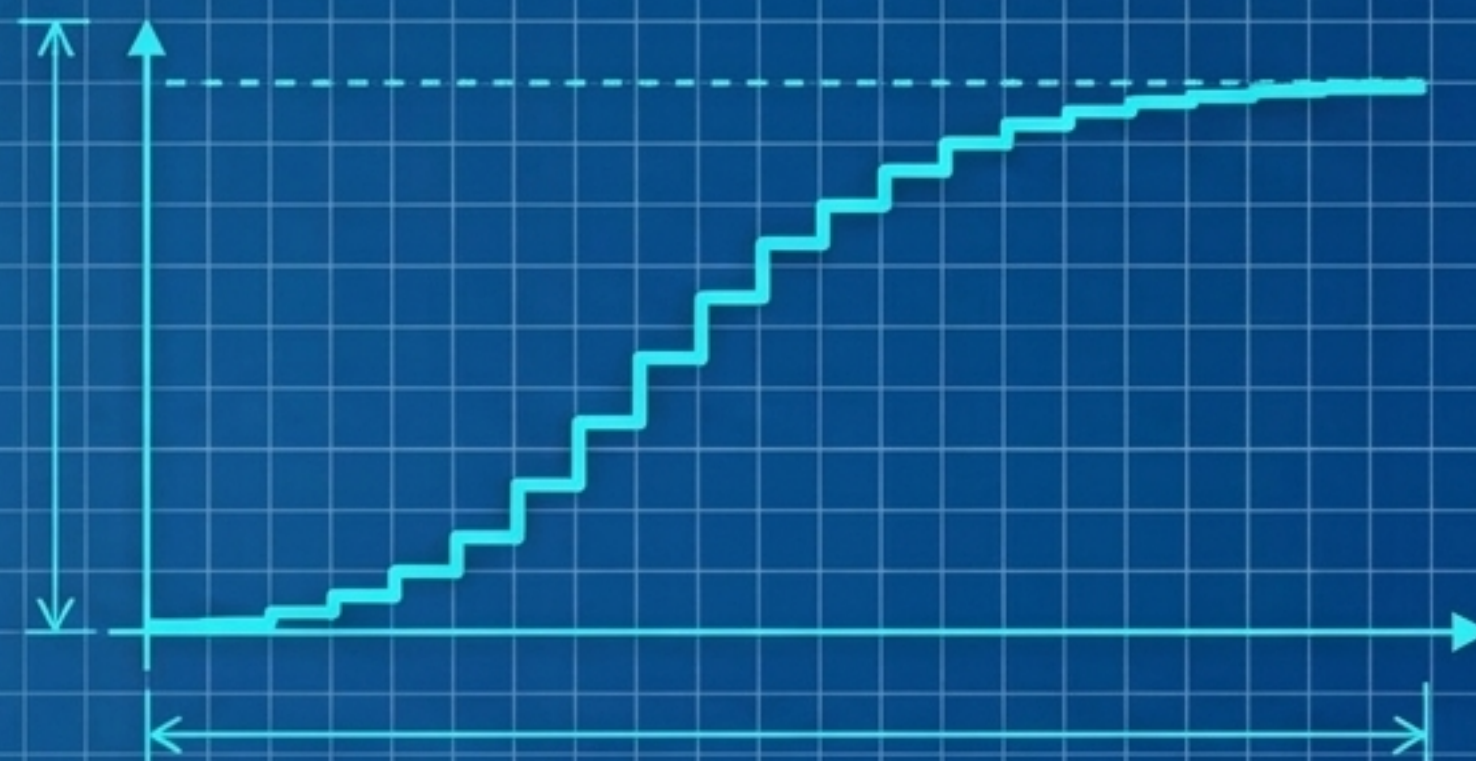
なぜ「徐々に」音量を変えるのか？

【×】不快な音量変化



急激な音量変化は耳に強い負担と不快感を与える。

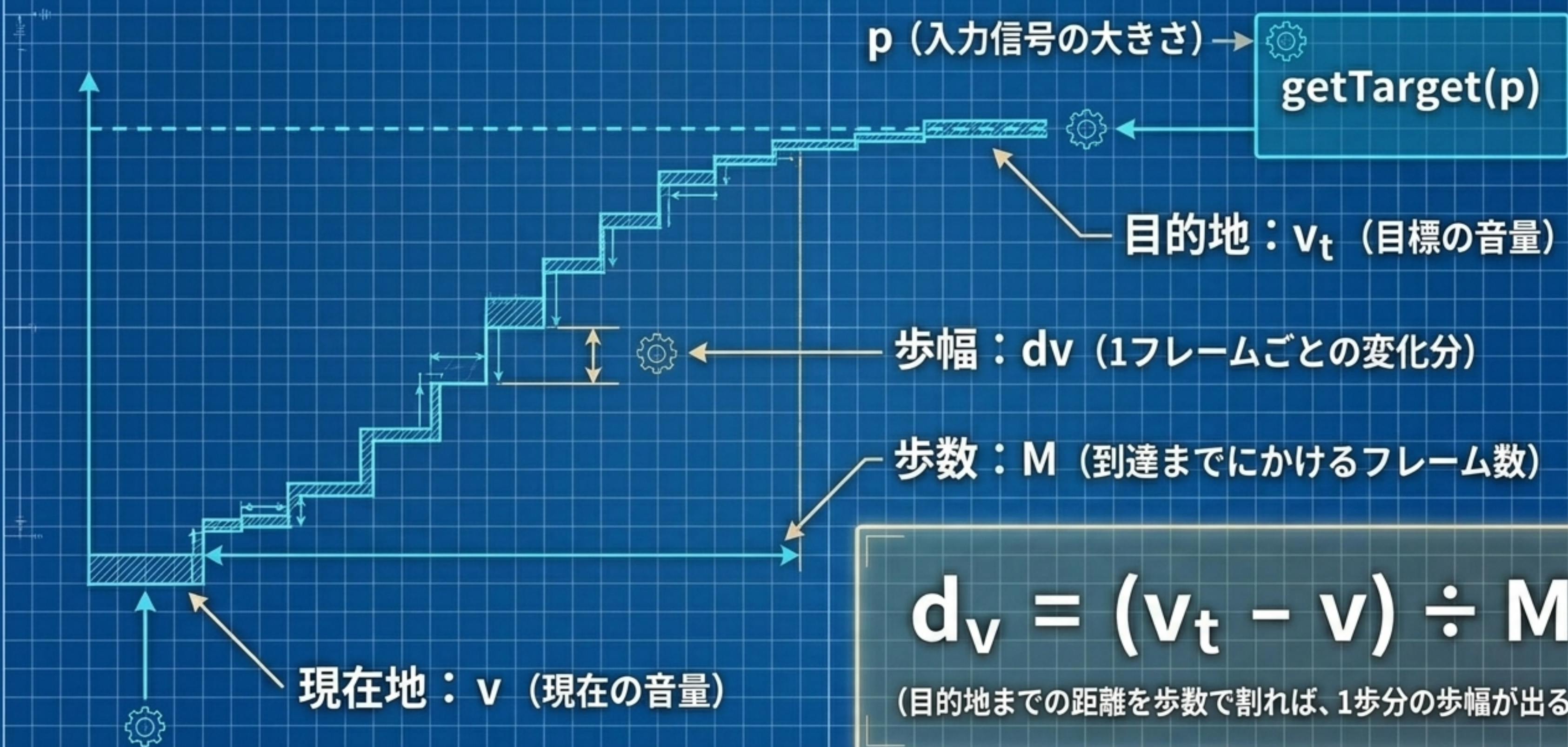
【○】AVCによる滑らかな音量変化



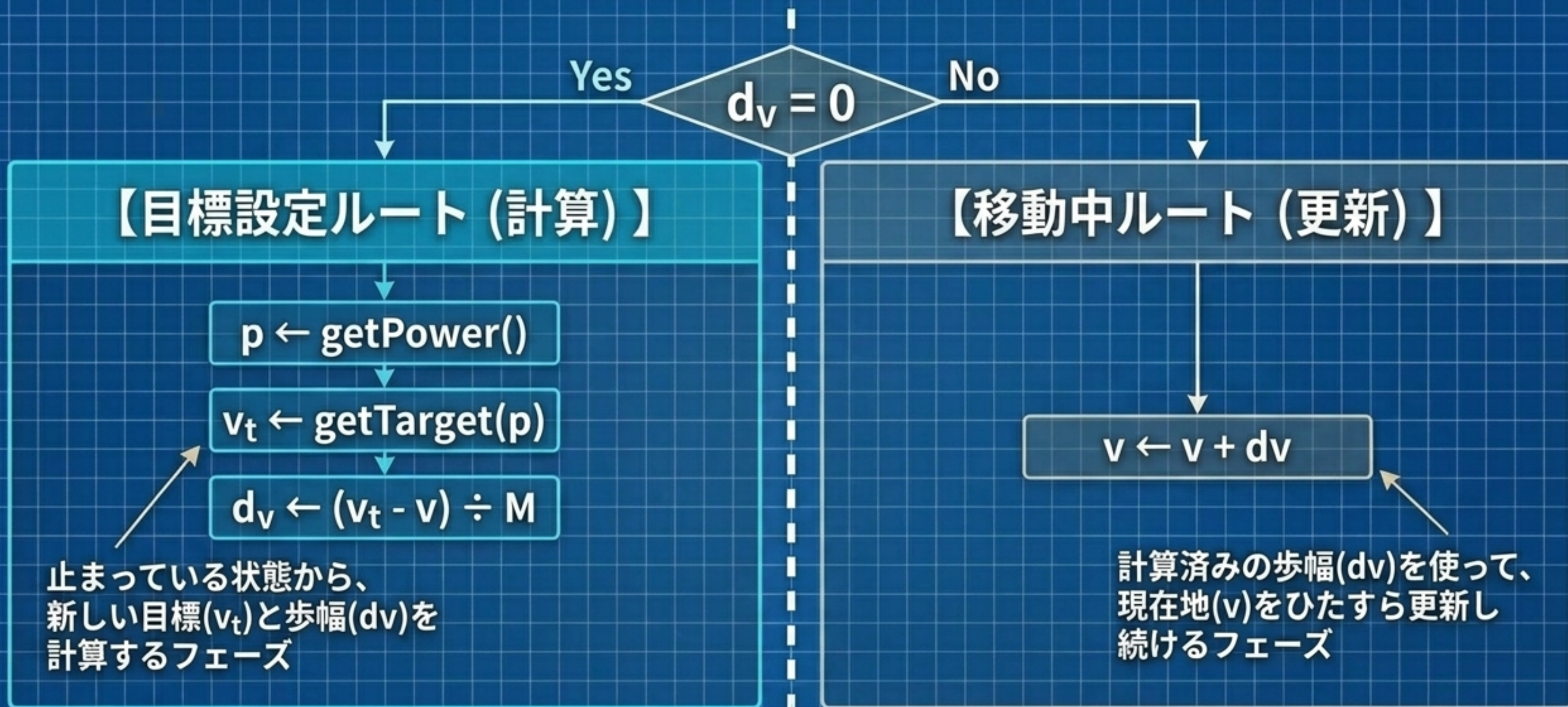
Mフレーム（複数回）に分けて、なだらかな階段状に目標へ近づけていく。

この「なだらかな階段を作る」ことこそが、今回のアルゴリズムに課せられた最大の目的です。

アルゴリズムを駆動する「現在地」と「目的地」



ルート分岐：今は「目標を設定する時」か「移動中」か？

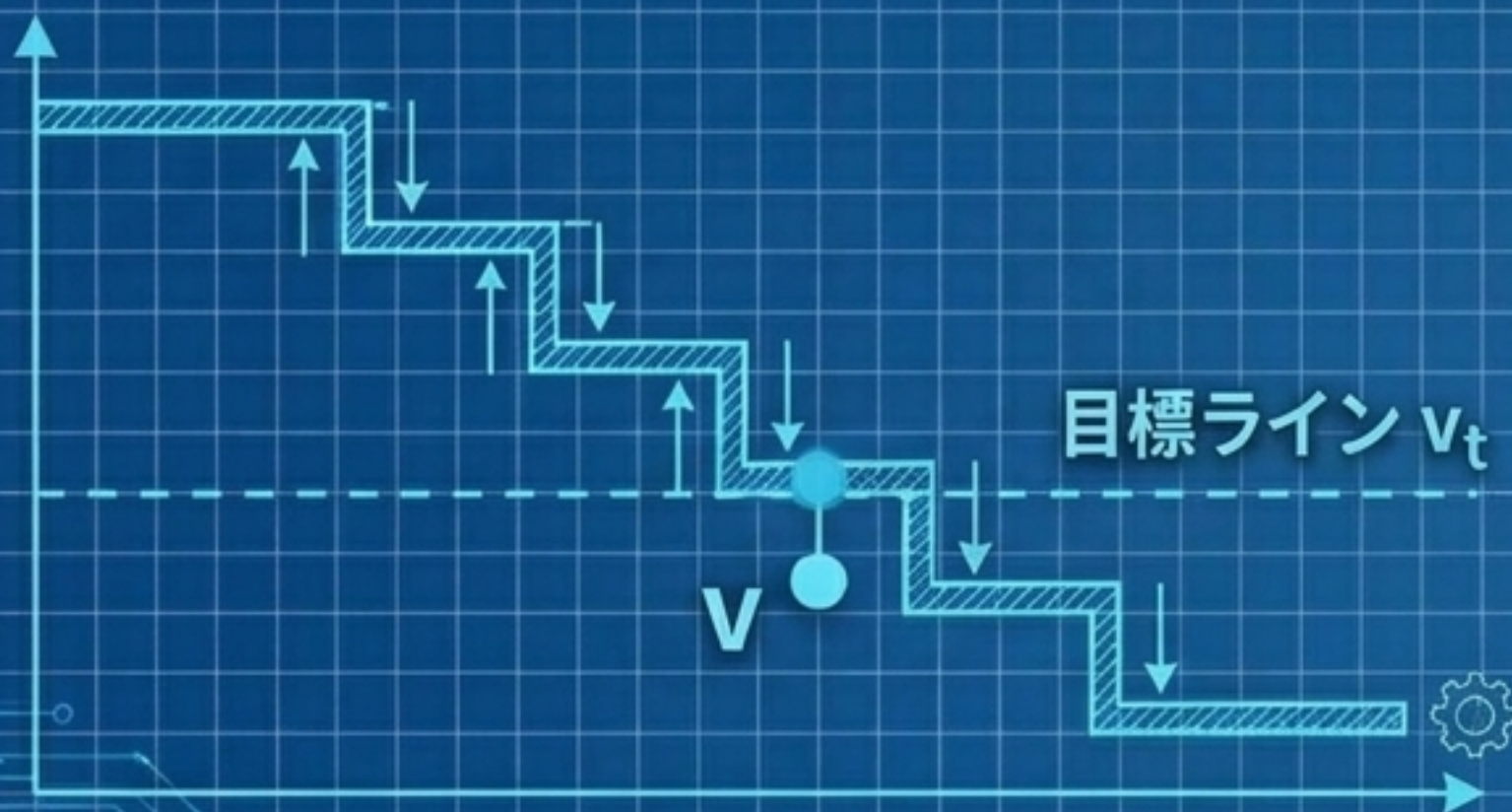


複雑に見えるフローチャートも、本質は入り口の判定による2つの役割分担に過ぎません。

[ア]のミッション：目的地を通り過ぎていないか？

【音量を下げている場合】 $d_v < 0$

【音量を上げている場合】 $d_v > 0$



到着条件： $v \leq v_t$

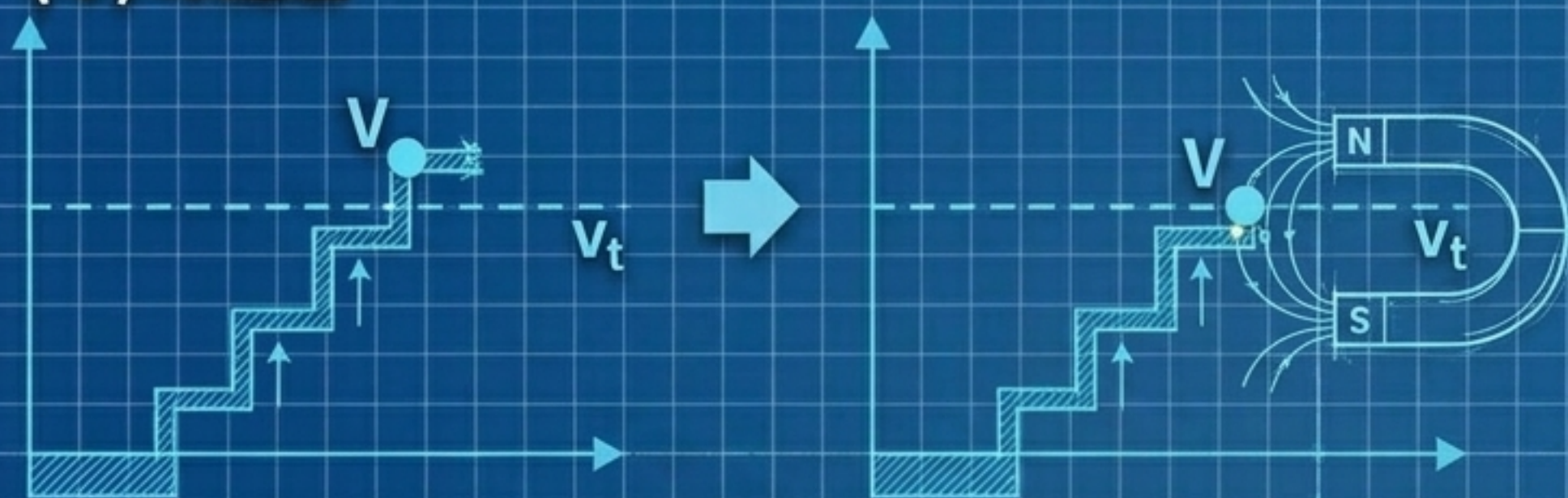
(現在地 v が目標ライン v_t を下回った瞬間)

【ア】に入るべき到着条件は？

(現在地 v が目標ライン v_t を「突き抜けた」
ことをどう数式で表現するか？)

[イ][ウ]のミッション：目的地に着いた後の「後処理」

(イ) の処理



(イ) 現在地 v の補正

割り算の都合上、最後の更新で v は目標値 v_t をわずかに飛び越える可能性があります。ピッタリ合わせるには v に何を代入して上書きすべきですか？

$v \leftarrow [\text{イ}]$

(ウ) の処理



歩幅(dv)メーター



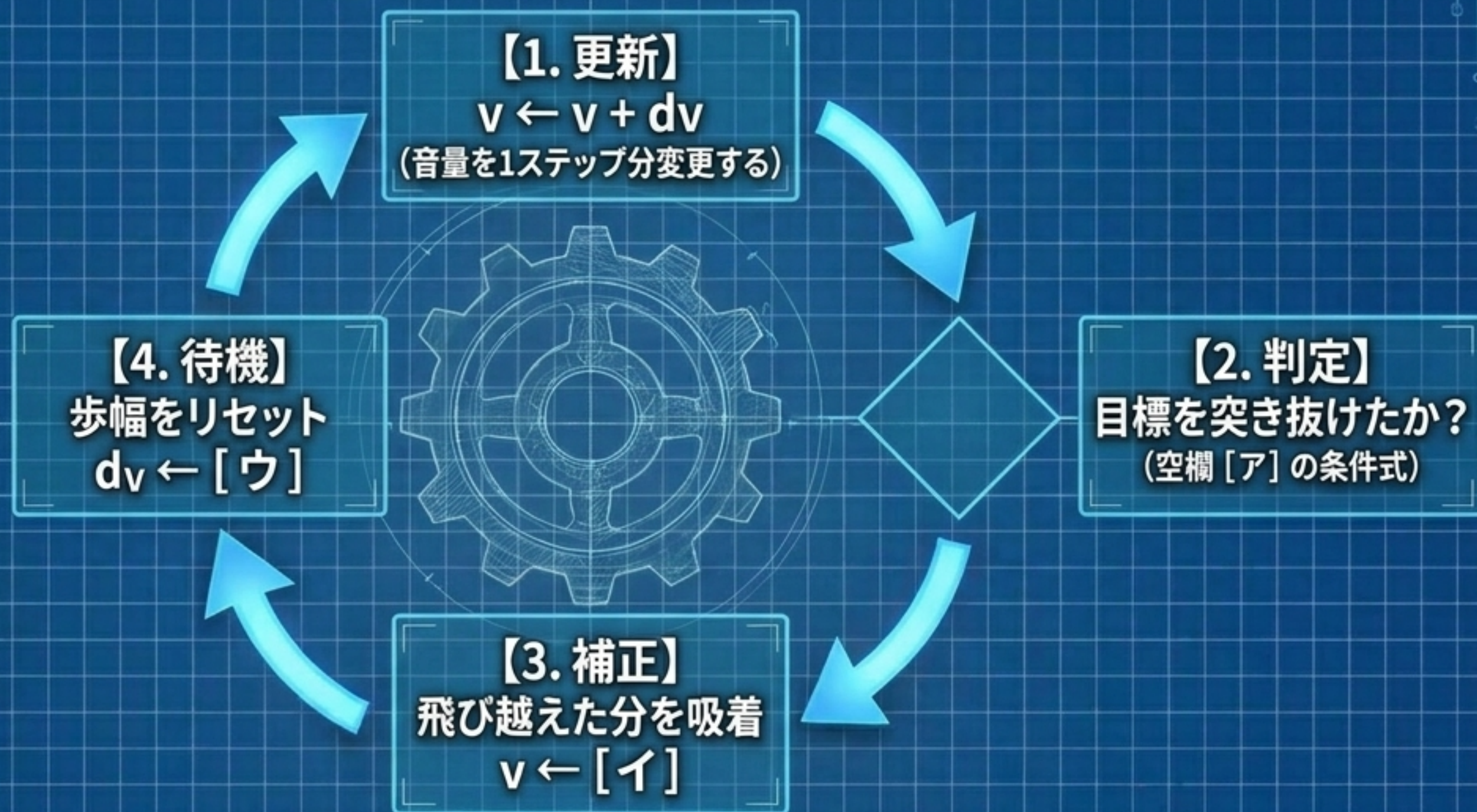
歩幅(dv)メーター

(ウ) 歩幅 dv の初期化

目的地に着いたので、これ以上音量を変える必要はありません。次のフレームで確実に「目標設定ルート」に進ませるためには、歩幅 dv をどういう状態に戻すべきですか？

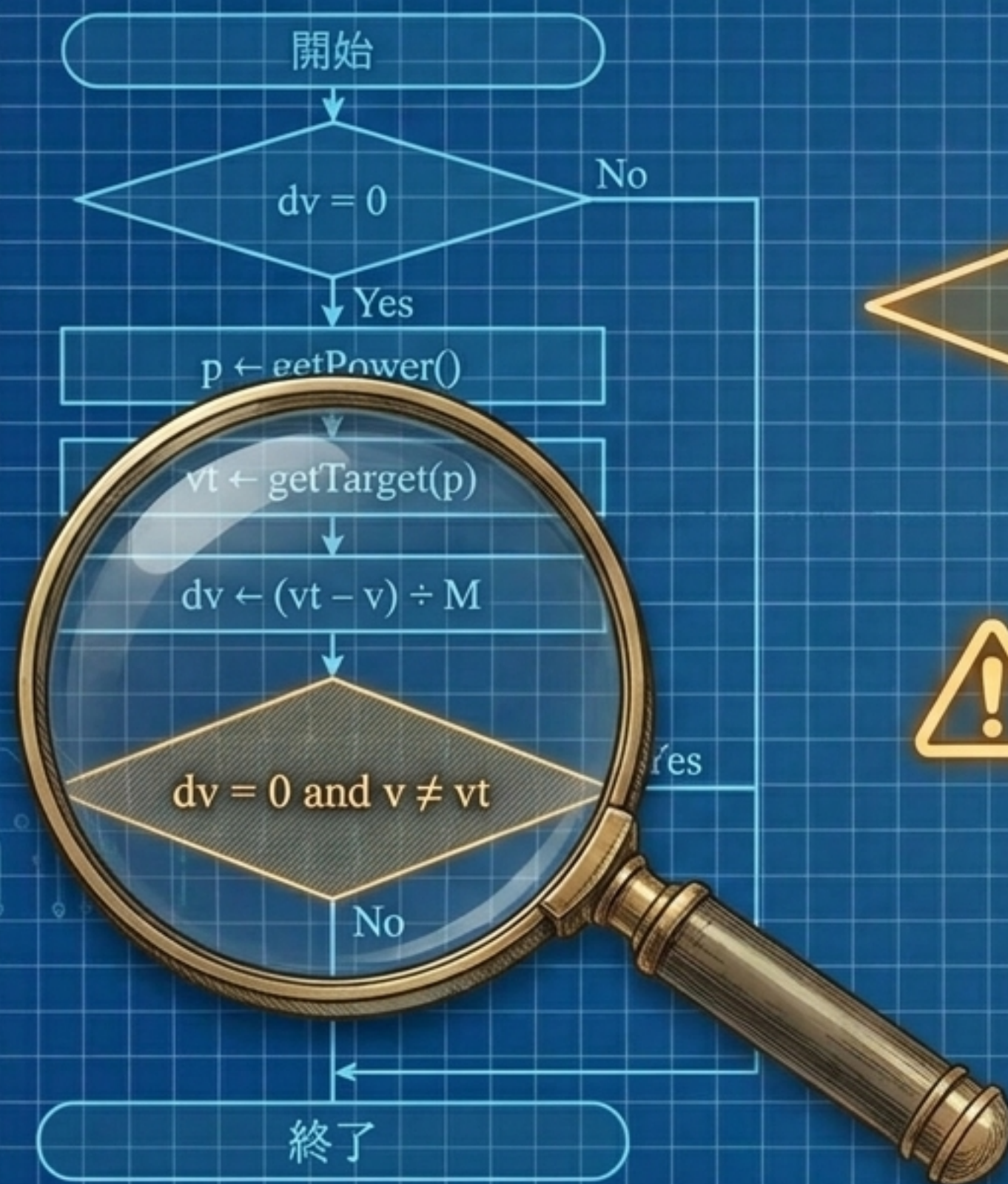
$dv \leftarrow [\text{ウ}]$

音量更新のワンサイクル（セルフチェック）

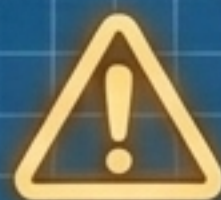


移動中ルートは、この4つのステップが1つのサイクルとして完結しています。
それぞれの空欄が持つ「役割」から、入るべき数式や変数が論理的に導き出せたはずです。

なぜ「直ちに音量を目標にする」例外ルートが存在するのか？



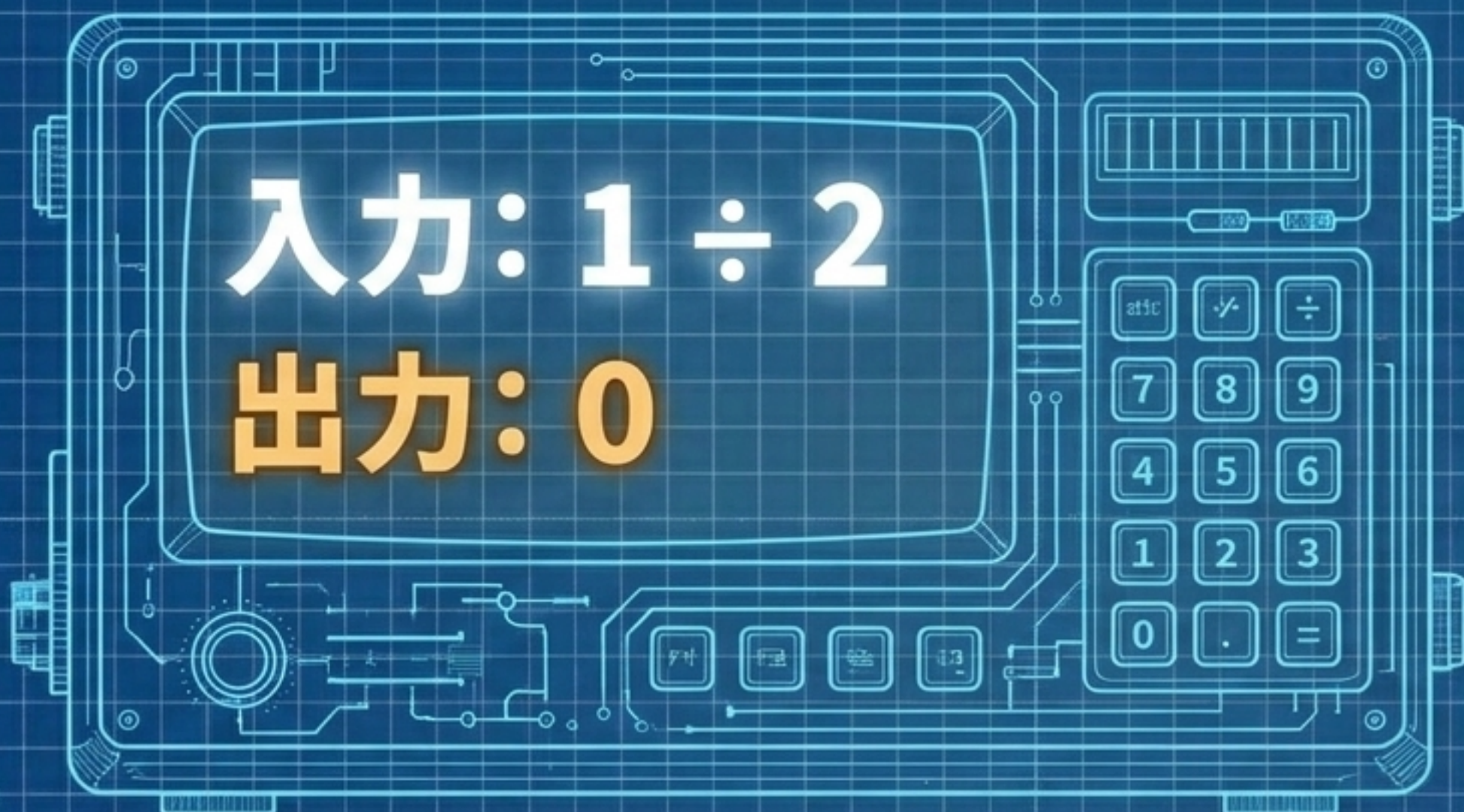
$dv = 0$ and $v \neq vt$



最大の謎：

- ① 直前の処理で $dv \leftarrow (vt - v) \div M$ と計算したばかりなのに、なぜ歩幅 dv が 0 になってしまうことがあるのか？
- ② そしてその時、現在地 v と目標 vt が一致していない ($v \neq vt$) のはなぜか？

コンピュータの世界の「割り算」の性質



【システムの絶対ルール】

演算は全て整数演算であり、浮動小数点演算は使用しない。

【シミュレーション】

・もし、目標音量(v_t)と現在の音量(v)の差が非常に小さかったら？

残り距離が「1」、かけるフレーム数(M)が「2」の場合、「 $1 \div 2$ 」の結果は小数点が切り捨てられ、無慈悲にも「0」となります。これが例外ルートに陥るカラクリです。

状態の言語化(40字以内の記述に向けて)

1. 状態

現在地(v)と
目的地(vt)の距離が
どうなっているか?
(遠い?近い?)

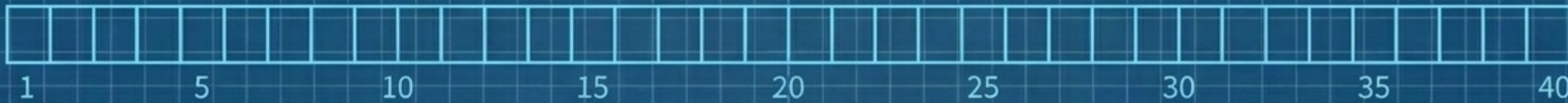
2. 現象

その結果、計算された
1フレームごとの
変化量(dv)はどう
なってしまったか?

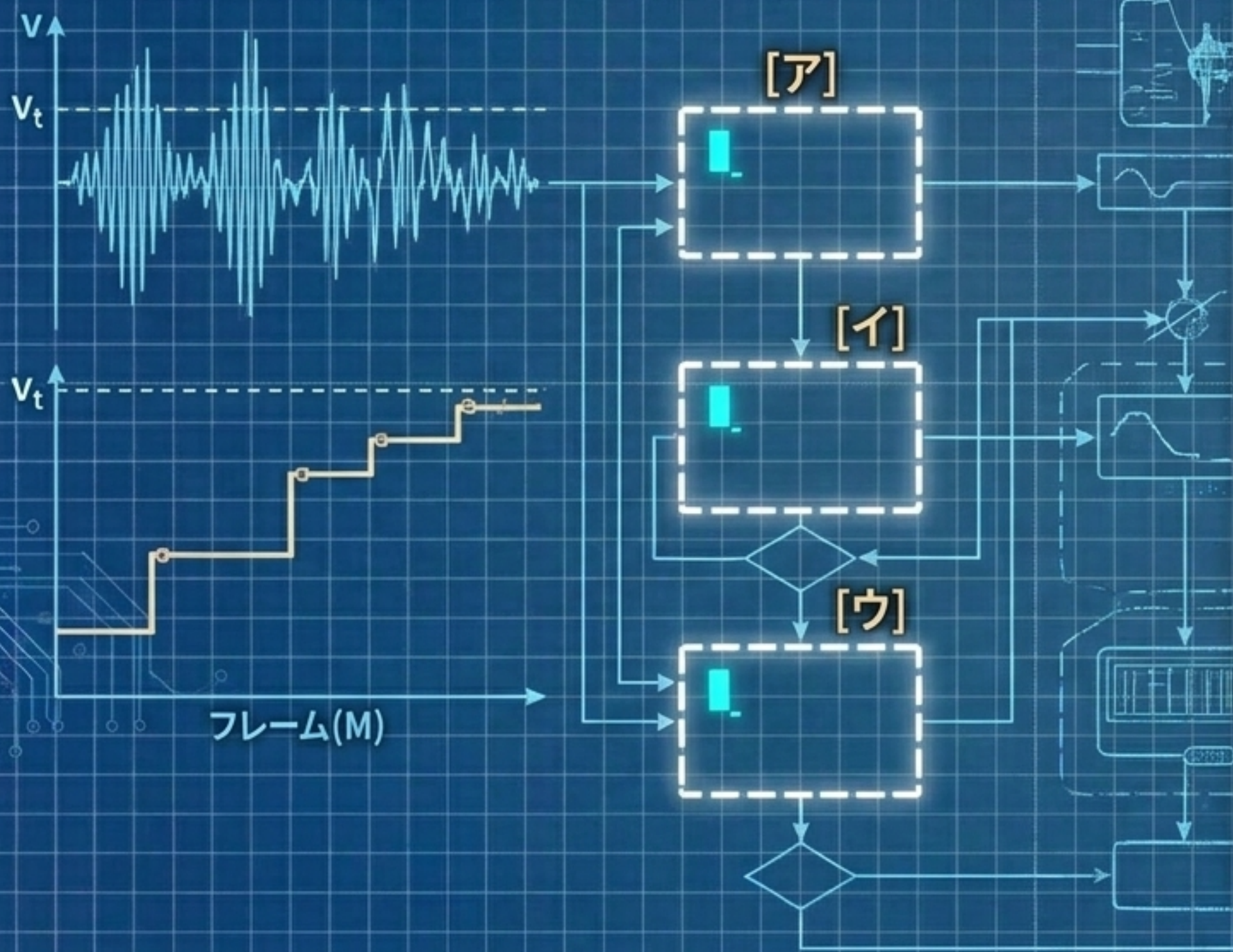
3. 結論

だから、徐々に
変えることを諦め、
音量をどうする必と
判断したのか?

これら3つの要素を繋ぎ合わせることで、
エンジニアの視点に立った解答が自然と浮かび上がります。



全てのヒントは出揃いました。あとは設計図を完成させるだけです。



アルゴリズムは暗記するものではなく、
目的を持った「設計」です。

- 変数たちが持つ物理的な役割
- フローチャートが担う2つのフェーズ
- ハードウェアの制約(整数演算)がもたらす影響

全体像が見えた今、これは単なる無機質な穴埋め問題ではなく、エンジニアが意図を持って仕掛けた「極めて合理的なロジック」として、あなたの目に映っているはずです。