PINN Phasor Fault Localization: Mathematical Derivation and Calibration

1 Scope

This document derives the method implemented in:

- `projects/pinn_phasor_localization/model.py`,
- `projects/pinn_phasor_localization/phasor.py`,
- `projects/pinn_phasor_localization/trainer.py`,
- `projects/pinn_phasor_localization/run.py`.

The pipeline has two layers:

1. **Physics inversion** from COMTRADE waveforms to $\hat{x}_f \in (0, 1)$ and Z_f.
2. **Optional supervised calibration** mapping $\hat{x}_f$ to distance labels.

2 Phasor telegrapher model with a localized shunt fault

Let $x \in [0, 1]$ be normalized line position ($x = 0$ left terminal, $x = 1$ right terminal). At fixed system frequency $\omega = 2\pi f$, define complex phasors

$$V(x) = V_r(x) + jV_i(x), \qquad I(x) = I_r(x) + jI_i(x). \tag{1}$$

The model is

$$\frac{dV}{dx} = -z_n I, \tag{2}$$

$$\frac{dI}{dx} = -y_n V - I_f\, G_\sigma(x - x_f), \tag{3}$$

with

$$z_n = R_n + jX_n, \qquad y_n = G_n + jB_n, \tag{4}$$

$$I_f = \frac{V(x_f)}{Z_f}, \qquad Z_f = Z_{f,r} + jZ_{f,x}. \tag{5}$$

G_σ is a Gaussian approximation of $\delta(x - x_f)$:

$$G_\sigma(\xi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\xi^2}{2\sigma^2}\right). \tag{6}$$

In code, $\sigma = 0.04$ in normalized line units.

2.1 Real-imaginary residual form

Using

$$z_n I = (R_n I_r - X_n I_i) + j(R_n I_i + X_n I_r), \tag{7}$$

$$y_n V = (G_n V_r - B_n V_i) + j(G_n V_i + B_n V_r), \tag{8}$$

equation (2) gives

$$\frac{dV_r}{dx} + R_n I_r - X_n I_i = 0, \tag{9}$$

$$\frac{dV_i}{dx} + R_n I_i + X_n I_r = 0. \tag{10}$$

Let $I_f = I_{f,r} + jI_{f,i}$. Equation (3) gives

$$\frac{dI_r}{dx} + G_n V_r - B_n V_i + I_{f,r} G_\sigma(x - x_f) = 0, \tag{11}$$

$$\frac{dI_i}{dx} + G_n V_i + B_n V_r + I_{f,i} G_\sigma(x - x_f) = 0. \tag{12}$$

These are exactly the four PDE residual channels minimized by the PINN.

3 Parameterization and constraints

The trainable raw parameters are transformed as:

$$x_f = \sigma(\texttt{raw\_xf}) \in (0, 1), \tag{13}$$

$$Z_{f,r} = \text{softplus}(\texttt{raw\_Zf\_r}) + \varepsilon > 0, \tag{14}$$

$$Z_{f,x} = \texttt{raw\_Zf\_x} \in \mathbb{R}, \tag{15}$$

$$R_n = \text{softplus}(\texttt{raw\_R}) > 0, \quad X_n = \text{softplus}(\texttt{raw\_X}) > 0, \tag{16}$$

$$G_n = \text{softplus}(\texttt{raw\_G}) > 0, \quad B_n = \text{softplus}(\texttt{raw\_B}) > 0. \tag{17}$$

By default, line parameters are frozen to physical priors:

$$R_n^0 = 0.0233178327,\ X_n^0 = 0.4224245054,\ G_n^0 = 3.0 \times 10^{-4},\ B_n^0 = 0.4224245054. \tag{18}$$

4 COMTRADE preprocessing and boundary phasors

4.1 Analog reconstruction

For each channel y:

$$y[n] = a_y\, y_{\text{raw}}[n] + b_y. \tag{19}$$

4.2 Fault inception and arrival picks

Combined current magnitude:

$$I_{\text{mag}}(t) = \sqrt{\sum_{p \in \{L,R\}} \sum_{\phi \in \{1,2,3\}} I_{p,\phi}(t)^2}. \tag{20}$$

Fault inception t_0 is detected from a smoothed derivative threshold:

$$\left| \frac{d}{dt} \text{MA}(I_{\text{mag}}) \right| > \mu + 8\sigma. \tag{21}$$

Arrival times per terminal use a local derivative threshold:

$$t_L : \left| \frac{dI_L^{\text{abs}}}{dt} \right| > \mu_L + 6\sigma_L, \qquad t_R : \left| \frac{dI_R^{\text{abs}}}{dt} \right| > \mu_R + 6\sigma_R, \tag{22}$$

with sub-sample interpolation at first crossing.

4.3 Arrival-time geometry prior

Using one-way travel identity

$$t_L = t_0 + x_f \tau, \qquad t_R = t_0 + (1 - x_f)\tau, \tag{23}$$

we obtain

$$\tau_{\text{arr}} = t_L + t_R - 2t_0, \tag{24}$$
$$x_{\text{arr}} = \frac{1}{2} + \frac{t_L - t_R}{2\tau_{\text{arr}}}. \tag{25}$$

If $\tau_{\text{arr}} \leq 0$, the prior is treated as invalid.

4.4 Multi-window phasor extraction

For window $k = 0, \ldots, K - 1$:

$$t_k^{(s)} = t_0 + (\texttt{skip\_cycles} + k\, \texttt{stride\_cycles}) T_0, \tag{26}$$
$$t_k^{(e)} = t_k^{(s)} + \texttt{post\_cycles}\, T_0, \qquad T_0 = \frac{1}{f}. \tag{27}$$

For any signal $s(t)$ in the window:

$$\hat{s} = 2\, \overline{s(t) e^{-j\omega t}} = 2 \left(\overline{s \cos \omega t} - j \overline{s \sin \omega t} \right). \tag{28}$$

Apply this to phase voltages/currents at both terminals.

4.5 Sequence decomposition and fault-type weighting

For phase phasors (P_a, P_b, P_c), with $a = e^{j2\pi/3}$:

$$P_0 = \frac{P_a + P_b + P_c}{3}, \tag{29}$$
$$P_1 = \frac{P_a + aP_b + a^2 P_c}{3}, \tag{30}$$
$$P_2 = \frac{P_a + a^2 P_b + aP_c}{3}. \tag{31}$$

Weighted equivalent (fault-type dependent (w_0, w_1, w_2)):

$$P_{\text{eq}} = w_0 P_0 + w_1 P_1 + w_2 P_2. \tag{32}$$

Right-terminal current sign convention is unified as line-forward:

$$I_R^{\text{eq}} \leftarrow -I_R^{\text{eq}}. \tag{33}$$

4.6 Normalization

Across windows:

$$v_{\text{scale}} = \max\left(\max_k |V_{L,k}|, \max_k |V_{R,k}|, 1\right), \tag{34}$$

$$i_{\text{scale}} = \max\left(\max_k |I_{L,k}|, \max_k |I_{R,k}|, 1\right). \tag{35}$$

Boundary phasors used by training are normalized:

$$\tilde{V}_{(\cdot),k} = \frac{V_{(\cdot),k}}{v_{\text{scale}}}, \qquad \tilde{I}_{(\cdot),k} = \frac{I_{(\cdot),k}}{i_{\text{scale}}}. \tag{36}$$

5 PINN objective

Let K be number of valid windows and N_c collocation points.

5.1 Boundary loss (multi-window)

$$\begin{aligned}\mathcal{L}_{\text{bc}} = \frac{1}{K}\sum_{k=1}^{K}\Big(&|V_r(0) - \tilde{V}_{L,r}^{(k)}|^2 + |V_i(0) - \tilde{V}_{L,i}^{(k)}|^2 + |I_r(0) - \tilde{I}_{L,r}^{(k)}|^2 + |I_i(0) - \tilde{I}_{L,i}^{(k)}|^2 \\ &+|V_r(1) - \tilde{V}_{R,r}^{(k)}|^2 + |V_i(1) - \tilde{V}_{R,i}^{(k)}|^2 + |I_r(1) - \tilde{I}_{R,r}^{(k)}|^2 + |I_i(1) - \tilde{I}_{R,i}^{(k)}|^2\Big).\end{aligned} \tag{37}$$

5.2 PDE loss

With residuals (9)–(12) at random $x_m \sim U(0,1)$:

$$\mathcal{L}_{\text{pde}} = \frac{1}{N_c}\sum_{m=1}^{N_c}\left(r_1(x_m)^2 + r_2(x_m)^2 + r_3(x_m)^2 + r_4(x_m)^2\right). \tag{38}$$

5.3 Arrival anchor loss

If x_{arr} from (25) is valid:

$$\mathcal{L}_{x\text{arr}} = (x_f - x_{\text{arr}})^2. \tag{39}$$

Otherwise $\mathcal{L}_{x\text{arr}} = 0$.

5.4 Line-prior regularization

When line parameters are trainable:

$$\mathcal{L}_{\text{line}} = (R_n - R_n^0)^2 + (X_n - X_n^0)^2 + (G_n - G_n^0)^2 + (B_n - B_n^0)^2. \tag{40}$$

If line parameters are frozen, $\mathcal{L}_{\text{line}} = 0$.

5.5 Total loss

$$\mathcal{L} = w_{\text{bc}}\mathcal{L}_{\text{bc}} + w_{\text{pde}}\mathcal{L}_{\text{pde}} + w_{x\text{arr}}\mathcal{L}_{x\text{arr}} + w_{\text{line}}\mathcal{L}_{\text{line}}. \tag{41}$$

6 Optimization schedule

Warmup (Adam, low lr). Optimize boundary + arrival (+ line reg if enabled), effectively without PDE.

Main Adam stage. Optimize full weighted objective.

LBFGS stage. Deterministic refinement with fixed collocation samples.

7 Recovered physical quantities

Predicted distance for known line length L_{km}:

$$\hat{d}_{\text{km}} = x_f \, L_{\text{km}}. \tag{42}$$

Fault impedance in ohms is obtained by undoing normalization:

$$Z_0 = \frac{v_{\text{scale}}}{i_{\text{scale}}}, \qquad \hat{Z}_{f,\Omega} = (Z_{f,r} + jZ_{f,x})\, Z_0. \tag{43}$$

8 Affine calibration layer

8.1 Need for calibration

If inversion output x_f is monotone but biased versus labels, use a global affine calibrator.

8.2 Model

Let labeled samples be (x_j, y_j) with

$$x_j = \hat{x}_{f,j}, \qquad y_j = \frac{FD_j}{L_{\text{km}}}. \tag{44}$$

Fit

$$\hat{y}_j = ax_j + b, \qquad (a, b) = \arg\min_{a,b} \sum_{j=1}^{N} (ax_j + b - y_j)^2. \tag{45}$$

Matrix form:

$$\boldsymbol{\theta} = \begin{bmatrix} a \\ b \end{bmatrix} = (X^\top X)^{-1} X^\top \boldsymbol{y}, \qquad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}. \tag{46}$$

Calibrated prediction:

$$x_j^{\text{cal}} = \text{clip}(ax_j + b, 0, 1), \qquad \hat{d}_{j,\text{km}}^{\text{cal}} = x_j^{\text{cal}} L_{\text{km}}. \tag{47}$$

8.3 Leave-one-out (LOO) estimate

For each j, fit $(a^{(-j)}, b^{(-j)})$ on all samples except j:

$$x_{j,\text{loo}}^{\text{cal}} = \text{clip}\left(a^{(-j)} x_j + b^{(-j)}, 0, 1\right), \qquad \hat{d}_{j,\text{km}}^{\text{cal,loo}} = x_{j,\text{loo}}^{\text{cal}} L_{\text{km}}. \tag{48}$$

LOO gives a less optimistic estimate than fitting and testing on the same set.

9 Metrics

For residuals $e_j = \hat{d}_j - d_j$:

$$\mathrm{MAE} = \frac{1}{N} \sum_j |e_j|, \tag{49}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum_j e_j^2}, \tag{50}$$

$$\mathrm{Bias} = \frac{1}{N} \sum_j e_j, \tag{51}$$

$$\mathrm{MaxAbs} = \max_j |e_j|. \tag{52}$$

The same definitions are used for raw, calibrated, and LOO-calibrated predictions.

10 Interpretation

The pipeline is intentionally hybrid:

1. physics-constrained inversion gives structured latent estimates (x_f, Z_f),
2. calibration corrects systematic mapping bias from latent coordinate to distance.

Hence, calibration improves absolute distance accuracy but does not replace the physics layer.