State and Parameter Estimation with Uncertainty Quantification Using GANs and Markov Chain Monte Carlo Methods

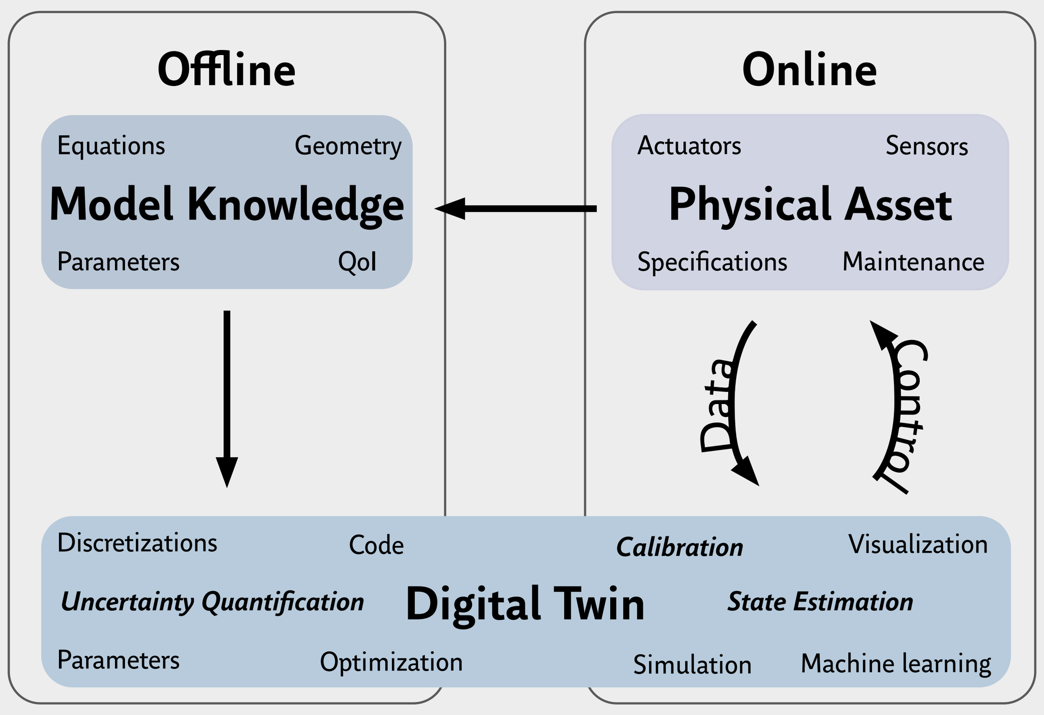
Nikolaj T. Mücke, Benjamin Sanderse, Sander Bohté, Cornelis Oosterlee

Enabling Digital Twins with Real-Time Calibration

A digital twin is a computational model that is continuously updated in order to represent a physical asset. It consists of a mix of *computational models* derived from domain knowledge and *data-driven models* trained on observation

The usages of a digital twin ranges from *forecasting* and *control*, to *monitoring* and *fault detection*.

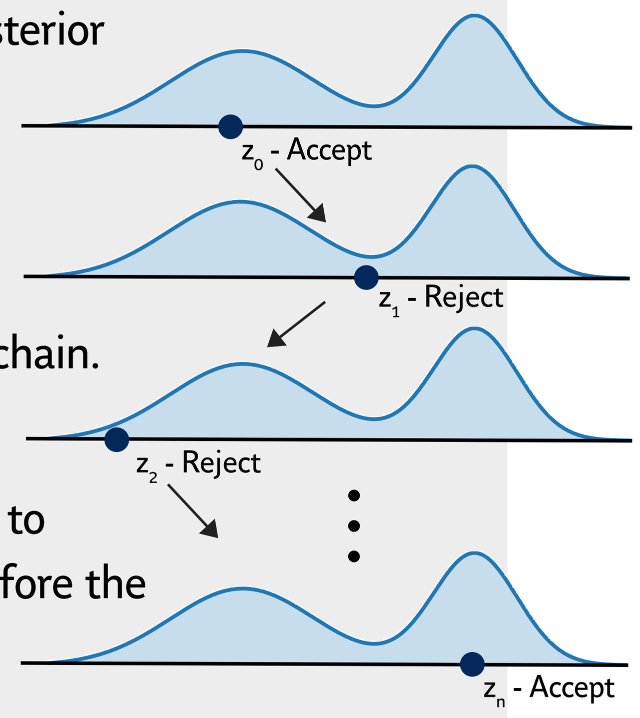
In order for a digital twin to mirror the physical asset at all times, ***real-time calibration*** against sensor data is undeniably *indispensable*.



Markov Chain Monte Carlo

One can sample from the posterior by creating a Markov chain with the posterior as the equilibrium distribution. New samples are obtained by computing new states of the chain.

While MCMC methods are efficient, it is often necesarry to compute **50,000+** samples before the chain has converged!



Bayesian Inverse Problems

Let $\mathbf{u} = (\mathbf{q}, \mathbf{m})$ be the state and model parameters. Then for given observations,

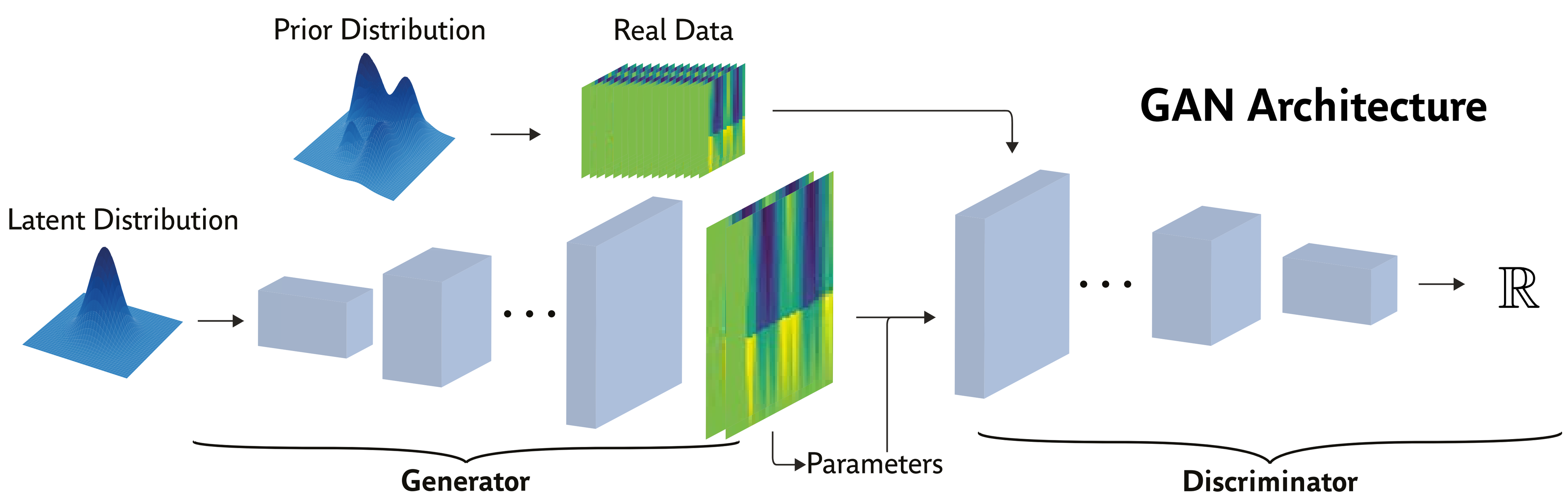
$$\underbrace{\mathbf{y}}_{\text{Observations}} = \underbrace{h(\mathbf{q})}_{\text{Observation operator}} + \eta, \quad \eta \sim \underbrace{\mu_\eta(\eta)}_{\text{Noise}},$$

The state is obtained by solving a PDE for a given set of parameters.

$$\underbrace{\rho_u(\mathbf{u}|\mathbf{y})}_{\text{Posterior}} = \frac{\overbrace{\rho_\eta(\mathbf{y} - h(\mathbf{u}))}^{\text{Likelihood}}\overbrace{\rho_u(\mathbf{u})}^{\text{Prior}}}{\underbrace{\int \rho_y(\mathbf{y}|\mathbf{u})\rho_u(\mathbf{u})\, \mathrm{d}\mathbf{u}}_{\text{Evidence}}}$$

Challenges

- *The posterior is intractable to compute*
- *The likelihood requires a forward solve of the (PDE) model*
- *The evidence is a high-dimensional integral*
- *Choosing a suitable prior is not easy*
- *... estimating the posterior is expensive!*



Wasserstein GANs

Approximates high-dimensional distribution by a low-dimensional latent distribution and a push forward map. The training is a "game" between a generator and a discriminator.

Generator. Approximates the real distribution by pushing forward a latent space distribution. Trained using:

$$L_G = -E_{\mathbf{z}\sim\mu_z}\left[D_\omega(G_\theta(\mathbf{z}))\right]$$

Discriminator. Discriminates between real and generated samples by assigning a real value to each sample. Trained using:

$$L_D = -E_{\boldsymbol{x}\sim\mu_r}\left[D_\omega(\boldsymbol{x})\right] + E_{\mathbf{z}\sim\mu_z}\left[D_\omega(G_\theta(\mathbf{z}))\right]$$

Offline Stage

1. **Compute high-fidelity training data with PDE scheme**
2. **Train Wasserstein GAN to approximate prior**

Online Stage

1. **Compute MAP estimate using GAN**
2. **Use MCMC to sample from latent space posterior**
3. **Generate state and parameters from posterior latent space samples**
4. **Compute relevant statistics from posterior samples**

Latent Space Posterior

Using the generator, one can sample the latent space instead of sample the high-fidelity space:

$$\rho_z(\mathbf{z}|\mathbf{y}) = \frac{\rho_\eta(\mathbf{y} - h(G(\mathbf{z})))\rho_z(\mathbf{z})}{\int_{\mathbb{R}^{N_z}} \rho_y(\mathbf{y}|\mathbf{z})\rho_z(\mathbf{z})\, \mathrm{d}\mathbf{z}}$$

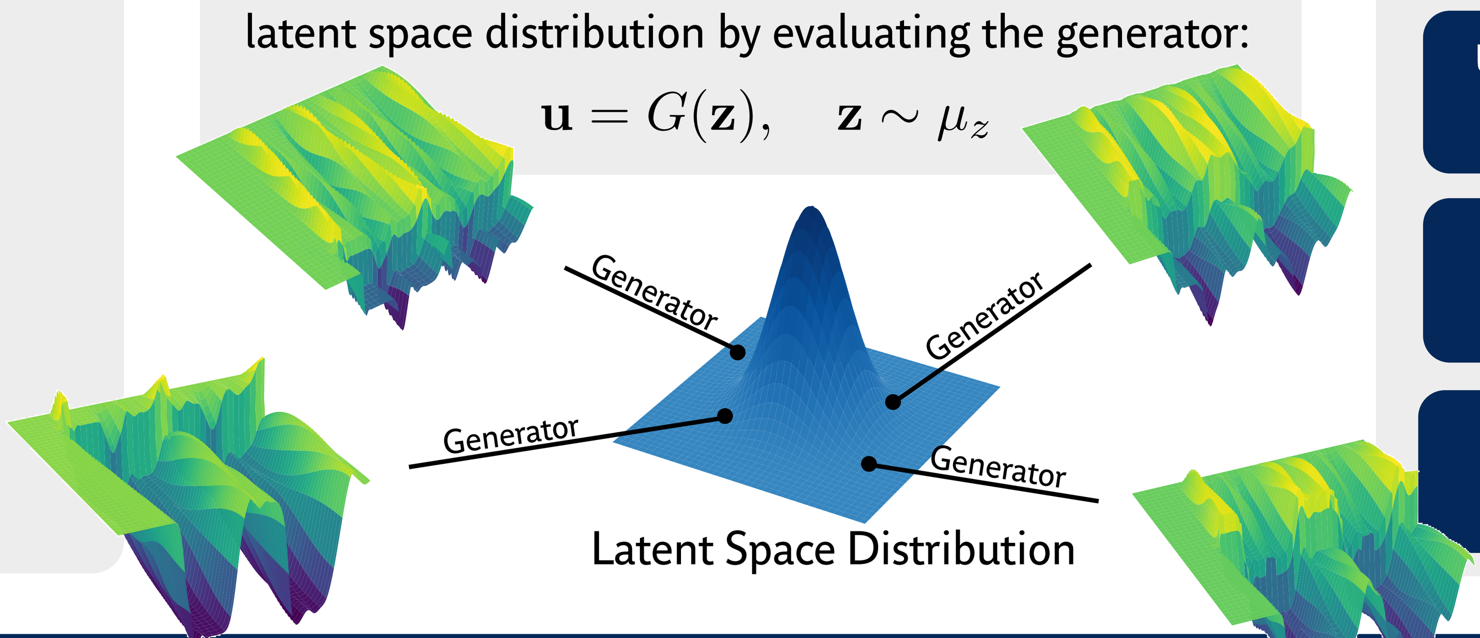
Advantages

- *The latent space is of much lower dimension*
- *The generator replaces the forward problem*
- *No need to choose an appropriate prior*
- *... large posterior sampling speed-ups!*

Latent Space Sampling

Train the GAN to generate states and parameters. New states and parameters are then sampled from the latent space distribution by evaluating the generator:

$$\mathbf{u} = G(\mathbf{z}), \quad \mathbf{z} \sim \mu_z$$



Test Case: Leakage Detection in Pipe Flow

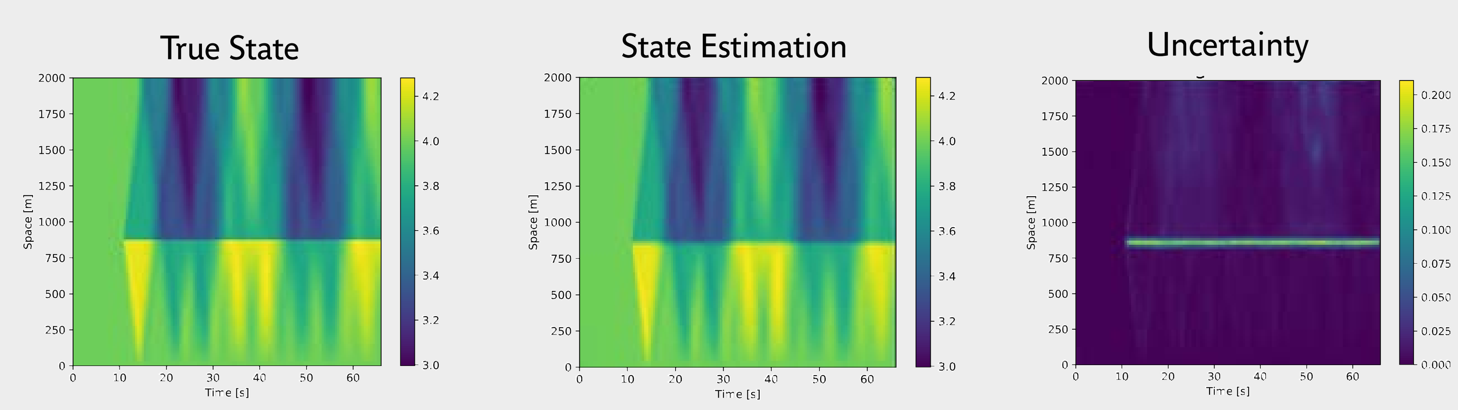
The equations describing pipe flow are a set of PDEs, consisting of a **mass conservation** equation:

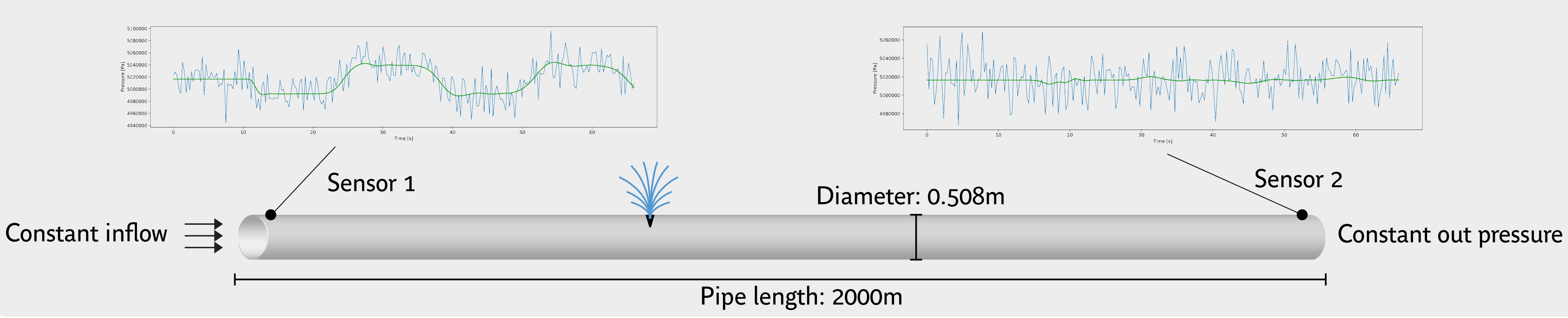
$$\partial_t q_1 + \partial_x(q_2) = \underbrace{C_d}_{\text{Discharge coefficient}}\sqrt{\rho(p(\rho) - p_{amb})}\,\delta(x - \underbrace{x_l}_{\text{Leakage location}})H(t - t_l)$$

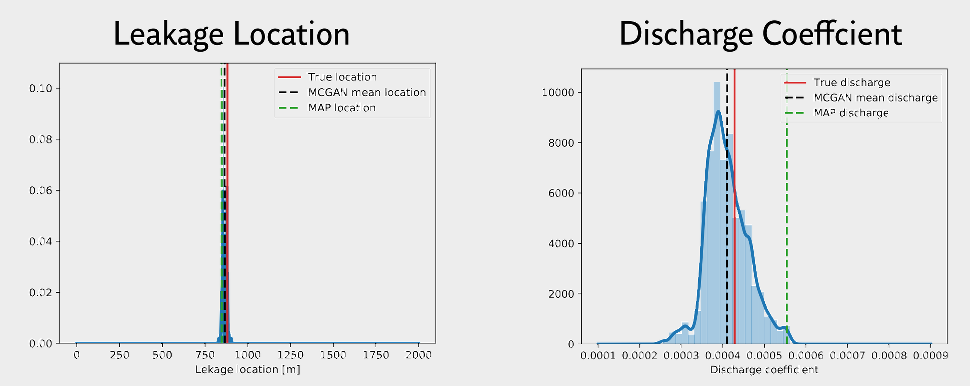
and **momentum conservation** equation:

$$\partial_t q_2 + \partial_x\left(\frac{q_2^2}{q_1} + p(\rho)A\right) = -\frac{D\pi}{2}f_f(q)\rho u^2$$

Goal: estimate the *leakage location* and *discharge coefficient* from noisy measurements.







For more information, contact: nikolaj.mucke@cwi.nl