Tennessee Eastman Process Simulation Dataset (TEPS)

The **Tennessee Eastman Process (TEP)** is a benchmark chemical process introduced by Downs and Vogel (1993) for testing fault detection and diagnosis algorithms. The **TEP Simulation Dataset** is widely used in **multivariate statistical process control (MSPC)** and **machine learning** to evaluate algorithms for anomaly detection, control charting, fault diagnosis, and prediction.

---

1. **Process Overview**

- A simulated chemical process producing two products (G and H) from four reactants (A, C, D, E).
- The process includes **reactors**, **separators**, **compressors**, and **cooling units**.
- Operated under **closed-loop control** using a hierarchical control strategy.
- Contains both **normal operating conditions** and **faulty scenarios**.

Absolutely — here is a **technical yet concise explanation** of the entire **Tennessee Eastman Process (TEP)**, including its steps, purpose, inputs, and flow.

Tennessee Eastman Process – Overview

Goal:

Produce chemical products **G** and **H** from gaseous reactants **A, C, D**, using **E** as a side reactant, with **B** as an inert component. Simultaneous reactions create **G** and **H**, but also undesired **by-product F**, requiring precise **control and separation**.

---

1. Inputs

Component	Role
A	Main reactant
C	Reactant for G formation
D	Reactant for H formation
E	Reacts with G to form F
B	Inert (non-reactive)

---

2. Step-by-Step Process

Step 1: Reactor (Vapor-Phase Reactions)

- Inputs: A, C, D, E, B

- Chemical Reactions:

  1. A + C → G (desired)
  2. A + D → H (desired)
  3. E + G → F (undesired)
  4. A → F (undesired)
  5. C + D → F (undesired)

- **Output**: Mixed vapor stream of G, H, F, unreacted A, C, D, E, and B

Diagram:

```
           +------------------+
A,C,D,E,B ->|     Reactor     |-> Reaction effluent (gas mixture)
           +------------------+
```

Step 2: Separation System

- **Condenser**: Cools reactor effluent → partial condensation

- **Separator**:

  - Vapor stream → recycled (mostly A, C, D, E)
  - Liquid stream → sent to stripper (mostly G, H, F, B)

Diagram:

```
           +-----------+
           | Condenser |
           +-----+-----+
                 |
           +-----v-----+
           | Separator |
           +-----+-----+
           |           |
       Recycle     Liquid to
       (A,C,D,E)     Stripper
```

Step 3: Stripper

- Injects steam at the bottom to strip out **light volatile** components.
- **Top**: F, B, some E (purged)
- **Bottom**: G, H (final product stream)

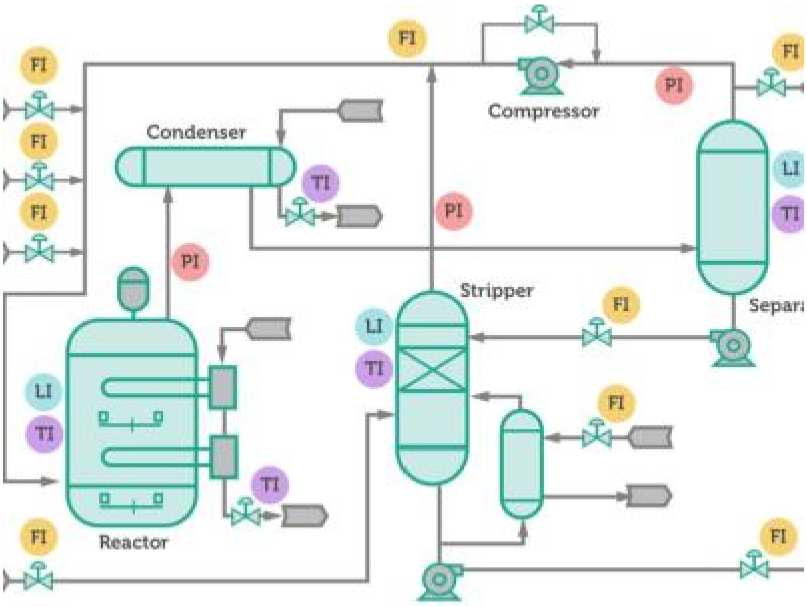
Diagram:

```
            +-----------+
            | Stripper  |
            +-----+-----+
                  |
          +--------v-------+
          | Product (G,H) |
          +---------------+
```

Step 4: Recycle & Purge

- **Recycle**: Unreacted A, C, D, E go back to the reactor
- **Purge**: Removes F and B to prevent buildup

Overall Material Flows



Key Engineering Challenges

- Nonlinear dynamics due to **recycling**
- **Faults**, e.g.:
  - Separator inefficiency
  - Reactor imbalance
  - Stripper underperformance
- Used widely in **process control and fault detection research**

---

2. **Dataset Characteristics**

2.1 Process Variables

- **52 variables in total:**
  - 41 measured variables (XMEAS$$ 1–41])
  - 11 manipulated variables (XMV$$ 1–11])

Each variable is a **time series**. Sampling is typically done every **3 minutes**.

2.2 Fault Modes

- 21 fault types (IDV 1–21), each representing a different kind of process disturbance.
- Faults range from:
  - Sensor drift/failure
  - Valve sticking
  - Process parameter change
  - Feed composition changes

Fault No	Description	Type
1	A/C feed ratio, B composition	Step change
2	B composition, constant	Step change
...	...	...
21	Random fault	Random

- Faults 1–20 have known sources; fault 21 is random and unspecified.

---

3. **Versions of the Dataset**

3.1 Original MATLAB/Simulink Simulation

- Provided by Downs & Vogel (1993), available from University of Tennessee.
- Generates raw data using a differential-algebraic model of the plant.

3.2 Pre-simulated Datasets

- CSV-format versions of TEPS.
- Each simulation:
  - Duration: 500–1000 samples (25–50 hours of simulated time)
  - Separate runs for each fault.
  - With and without injected faults.

3.3 Extended Versions

- Available in various research repositories.
- Include noise, delayed faults, multi-fault scenarios, missing data, etc.

---

4. Common Use Cases

Task	Description
Fault Detection	Identify onset of a fault using statistical or ML methods
Fault Classification	Determine fault type once detected
Dimensionality Reduction	Apply PCA, KPCA, Autoencoders to visualize or preprocess the data
Control Chart Evaluation	Test control chart methods like Hotelling's T^2, SPE, EWMA, etc.
Time Series Forecasting	Forecast XMEAS/XMV values for anomaly detection or predictive maintenance

5. Example Tools and Models Used

- **PCA**, **PLS**, **ICA**
- **Statistical Control Charts**: T^2, Q (SPE), EWMA, CUSUM
- **Neural Networks**: LSTM, Autoencoders, Variational AEs
- **Hybrid Approaches**: Combining ML with control theory

6. Example Preprocessing

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

df: pd.DataFrame = pd.read_csv("TEP_fault01.csv")
X: pd.DataFrame = df[[f"XMEAS({i})" for i in range(1, 42)] + [f"XMV({i})"
for i in range(1, 12)]]

scaler: StandardScaler = StandardScaler()
X_scaled: pd.DataFrame = pd.DataFrame(scaler.fit_transform(X),
columns=X.columns)
```

7. Sources

- Original paper: Downs, J. J., & Vogel, E. F. (1993). *A plant-wide industrial process control problem*. Computers & Chemical Engineering.
- TEPS datasets (MATLAB & CSV): available via UCI Repository and university FTP servers.