Development of the FASTPROOF project:
Implementation of a Modelica library for the simulation of offshore facilities

Presentation projet FASTPROOF, ENSTA MAE31
• FASTPROOF is a collaborative project between eni, saipem, eurobios and ENS Cachan

• The goal is to develop a software tool to simulate the operation of offshore oil facilities

• Focus is on open formats and open languages to share models and data - make them easy to extend

• The simulation concerns the global behavior of the system, with both physics and risk aspects
Two aspects:

• Physics
  ▶ Fluid flow in pipes
  ▶ Heat transfer: through pipe walls, from heating system, etc.
  ▶ Power consumption/energy conversion: electrical (generators), mechanical (pump), heat (heater)
  ▶ Controlled systems: check stability/efficiency of controllers

• Risk
  ▶ Failure of components
  ▶ Maintenance
  ▶ Reliability
Goals:

• Physics
  ▶ Help dimensioning some of the system characteristics, e.g. generator power, thickness of insulation layers, etc.
  ▶ Centralize the data to be shared between various actors
  ▶ Study the behavior of the whole system
  ▶ Based on the equations of physics

• Risk
  ▶ Estimate the availability of an installation
  ▶ Compare the reliability of two designs
  ▶ Assess the gain/risk ratio of an installation
  ▶ Stochastic simulation: probabilities, statistics
Framework design

• Physics
  ▶ Favor the *composability* of the developed models
  ▶ Ideally, the user can freely combine the models
  ▶ The components should be *re-usable*, i.e. not developed for a particular set of neighbor components
  ▶ The components should be *extensible*, i.e. easy to refine

• Risk
  ▶ Models easy to understand, with parameters that can actually be known (e.g. no conditional probabilities no one can know)
  ▶ *Synthetic output* to help the decision
Summary

1 Introduction

2 Physics simulation

3 Risk simulation

4 Two examples

5 Conclusion
- Modelica is a **simulation language**
  - It is **object-oriented** i.e. a new model can be built upon an existing model
  - It is **acausal**: no imposed causality (i.e. no pre-specified input/output)
- A model is a set of differential algebraic equations (+ algorithms)
- Models are connected through **connectors**
- The connectors automatically add the equations at the interface between components:
  - Potential variables: Equality at the connection point
  - Flow variables: sum to zero at the connection point
  - Stream variables: transported quantities (depends on the sign of an associated flow variable)
- Tools are available to compile and run models and plot the outputs (e.g. OpenModelica, Dymola)
Multi-physics simulation with Modelica

- Specific connectors are defined for each type of physical phenomenon
- Modelica allows the call to external code (e.g. in C language), which can be used to read parameters in a database for example
• Typical models in an offshore facility:

  ▶ Pipes: flow of a fluid mixture

  ▶ Heat: heat transfer through pipe wall, through heating material

  ▶ Electrical: diesel generator

  ▶ Controllers: heating system

  ▶ Other items: tanks, valves, junctions, pumps
Pipe model

• Connector for pipe models:
  ▶ Pressure $p$
  ▶ Mass flow rate $q_m$
  ▶ Vector of mass fractions $X = (X_1, X_2, \ldots, X_n)$
  ▶ Transported specific enthalpy $h$

• Equations:
  ▶ Volume conservation
  ▶ Mass balance
  ▶ Enthalpy balance
  ▶ Pressure drop
Pipe model

- Two ports: $a$ and $b$
- 0D model, quasi-stationary state
- Volume conservation:

$$\sum_{i=1}^{n} \frac{X_i^a q_m^a}{\rho_i} + \sum_{i=1}^{n} \frac{X_i^b q_m^b}{\rho_i} = 0$$

- Mass balance:

$$\dot{m}_i = X_i^a q_m^a + X_i^b q_m^b$$

- Enthalpy balance:

$$h = \sum_{i=1}^{n} X_i c_{pi} T$$

$$H = m_{total} h$$

$$H_{flow} = q_m^a h^a + q_m^b h^b$$

$$\dot{H} = H_{flow} + Q_{wall}$$
Pipe: friction model

• For one substance:

\[ dp_i = \text{pressureLossCoefficient}(\rho_i, D, L, \mu_i, v_i)|v_i| + dp_{gi} \]

• Velocity of substance \( i \) is

\[ v_i = \frac{X_i a q_m}{\rho_i A} \]

• Hydrostatic pressure term:

\[ dp_{gi} = \sin(\theta) \frac{m_i g}{A} \]

• \text{pressureLossCoefficient()} is given by:

\[ \text{pressureLossCoefficient}(\rho_i, v_i, D, L, \mu_i) = f_{\text{Darcy}}(Re_i, D) \frac{L \rho_i}{2D} \]

• Overall pressure drop:

\[ dp = \sum_{i=1}^{n} dp_i = p^a - p^b \]
Pipe model

- The pipe model is 0D: all the quantities are averaged inside the volume
- It is possible to approximate a 1D model by concatenating multiple 0D models
- Example: 1500m pipeline divided into 30 segments

![Graph showing temperature at the well, temperature along the pipeline, and sea temperature.](image)
Heat transfer models

- Heat connector: Temperature $T$, heat flow rate $Q_{flow}$
- Basic model of material with heat capacity (two heat connectors)

\[
\begin{align*}
C_p \frac{dT}{dt} &= Q^a_{flow} + Q^b_{flow} \\
Q^a_{flow} &= K_{th} (T^a - T) \\
Q^b_{flow} &= K_{th} (T - T^b)
\end{align*}
\]

- Insulation layers are stacks of blocks:
Electrical models

- Electric connector: electric potential \( v \), electric current \( i \)
- Example, diesel generator: two electrical connectors (\( e_1 \) and \( e_2 \)), one fluid connector

\[
\begin{align*}
    v_{e_1} - v_{e_2} &= V_0 \\
    i_{e_1} + i_{e_2} &= 0 \\
    q_m \times \text{specific energy}_\text{diesel} &= (v_{e_1} - v_{e_2}) i_{e_1}
\end{align*}
\]

- Heating system with temperature control: two electrical connectors (\( e_1 \) and \( e_2 \)), two heat connectors (\( h_1 \) and \( h_2 \)), one input signal (\( T_{\text{ref}} \))

\[
\begin{align*}
    Q &= (v_{e_1} - v_{e_2}) i_{e_1} \\
    i_{e_1} + i_{e_2} &= 0 \\
    C_p \frac{dT}{dt} &= Q + K_{\text{th}} (T_{h1} - T) + K_{\text{th}} (T_{h2} - T) \\
    Q &= \min(\max(K_p (T - T_{\text{ref}}), 0), Q_{\text{max}})
\end{align*}
\]
Summary

1 Introduction

2 Physics simulation

3 Risk simulation

4 Two examples

5 Conclusion
Two levels

• Macro level based on expert knowledge
  ▶ General description of the system with a small number of states

• Micro level based on statistics of observed failures
  ▶ The components are considered individually
• The system is modeled as a Markov chain
• Markov chains are sequences of random variables taking values in a so-called state space
• Monte-Carlo method: generate many random sequences of events
• Simulate decision strategies and context changes
• Results are reported using standard techniques in statistics:
  - Mean / variance representation
  - Assessment of extreme values
  - Assessment of resilience
  - Evaluation of confidence intervals
• The micro level is meant to model component failures and maintenance

• Based on statistics gathered on real installations

• Each item is considered independently from the others

• Data comes from the OREDA\textsuperscript{1} handbook: statistics for time-between-failures and time-to-repair

---
OREDA database

- Example of the values given by OREDA (centrifugal pump):

<table>
<thead>
<tr>
<th>Failure mode</th>
<th>Number of failures</th>
<th>Failure rate (per 10⁶ hours)</th>
<th>Repair time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>Critical</td>
<td>464</td>
<td>70.52</td>
<td>106.81</td>
</tr>
<tr>
<td>Degraded</td>
<td>537</td>
<td>237.3</td>
<td>267.91</td>
</tr>
<tr>
<td>Incipient</td>
<td>936</td>
<td>834.3</td>
<td>688.83</td>
</tr>
<tr>
<td>Unknown</td>
<td>12</td>
<td>4.5</td>
<td>6.65</td>
</tr>
</tbody>
</table>

- From these data, we can build a stochastic model of the component
Markov chain model (micro level)

- Continuous-time semi-Markov chain:

  - $p_i$: probability of transition to state $i$
  - $t_i$: holding time, i.e. time before the transition actually triggers
Markov chain model (micro level)

- Transition probabilities: 
  \[ p_{\text{failure } i} = \frac{\text{number of observed failures of type } i}{\text{number of observed failures of any kind}} \]

- Holding times are computed from failure rates (exponential distribution) and (min, mean, max) values (beta distribution)

![Diagram of Markov chain model](image)
• A component can have associated spare units

• Adding a spare to the less reliable component:

<table>
<thead>
<tr>
<th>Component</th>
<th>No spares</th>
<th>One spare generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>generator</td>
<td>97%</td>
<td>59%</td>
</tr>
<tr>
<td>diesel_pump</td>
<td>2.4%</td>
<td>3.1%</td>
</tr>
<tr>
<td>valve2</td>
<td>0.2%</td>
<td>1.6%</td>
</tr>
<tr>
<td>Failed</td>
<td>4.5%</td>
<td>1.1%</td>
</tr>
<tr>
<td>OK</td>
<td>96%</td>
<td>99%</td>
</tr>
</tbody>
</table>

Failure duration per component

Time spent in failed and normal states
Summary

1 Introduction

2 Physics simulation

3 Risk simulation

4 Two examples

5 Conclusion
Test cases

- We simulate the process of stopping the production for repair, waiting and restarting.
- Shutting down production causes the mixture to cool down while at rest in the line.
- The temperature in the line must not fall below a specific temperature...
- ... otherwise hydrates might form and block the line.
- Example of two strategies to prevent hydrate formation: **Hybrid loop** and **Heated line** designs.

![Diagram of production line with temperature labels]

- **Wellhead**: $T \approx 60^\circ C$
- **Deep sea**: $T \approx 5^\circ C$
- **Surface storage**
Test case 1

- Hybrid loop design: replace mixture in the production line with diesel
Test case 2

- Heated line design: heat the line with an electric heating system
- FPSO (Floating Production, Storage and Offloading) unit model
• Hybrid loop has five states: production, no-touch time, diesel injection, repair, recirculation → production restart

1) PRODUCTION

• Heated line has four states: production, no-touch time, repair, heating → production restart
System states

- Hybrid loop has five states: production, no-touch time, diesel injection, repair, recirculation → production restart

2) INJECTION

- Heated line has four states: production, no-touch time, repair, heating → production restart
System states

- Hybrid loop has five states: production, no-touch time, diesel injection, repair, recirculation → production restart

3) RECIRCULATION

- Heated line has four states: production, no-touch time, repair, heating → production restart
Example of outputs

- Hybrid loop
• Heated line
Example of outputs

- Comparison of diesel consumption
  - Hybrid loop
  - Heated line
Example of outputs of the physics simulation

- Comparison of power consumption: temperature-controlled heating system vs. open-loop heating system
Results of the stochastic simulation

• The physics simulation provides some of the values corresponding to each state: production volume, diesel consumption
• The values are used by the stochastic simulation to compute quantities of interest such as costs and gains
Example of output from Monte-Carlo simulation

Figure: Comparison of S-curves for 1000 Monte-Carlo runs
Conclusion

- A set of models to simulate simplified test-cases has been developed
- Using Modelica to describe the physical model makes their extension easy, even for non-programmers
- Risk simulation helps to compare the reliability of two designs
- The fluid model is still very simple: large place for improvement (dynamic model, phase change, energy exchange between phases)
- Can the micro and macro levels of the risk simulation be combined?