HIL TESTING OF A MODERN DRILLING RIG

Using Simulink as modelling tool
SOFTWARE IS FANTASTIC

Software allows us to design, build and operate amazing machines.

Software allows us to plan and conduct extremely complex missions.

Software has changed the way we live.
SOFTWARE IS EVERYWHERE

- **In your hand**
  - phones, pads, gadgets, …

- **In your home**
  - computers, kitchen appliances, stereos, TVs, smart devices, …

- **In critical infrastructure**
  - power distribution, water, sewage, banks, telecommunication, finance, health care, the internet, …

- **In transportation**
  - planes, cars, ships, …
-Eh, can you make it 3 feet wider?
Class Bridge

private double width = 60;
private double length = 4500;

public void tiltTowers45degrees();

- Sure, no problem. Change the length also?
SOFTWARE IS FRAGILE

```plaintext
L_M_BV_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BV) * G_M_INFO_DERIVE(T_ALG.E_BV));

if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;

L_M_BH_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BH) * G_M_INFO_DERIVE(T_ALG.E_BH));

if L_M_BH_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BH) := 16#7FFF#
elsif L_M_BH_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BH) := 16#8000#
else
  P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BH_32));
end if;
```
SOFTWARE ON A DRILLING RIG IS A CHALLENGE

- Crane control system
- Well control systems
- BOP control system
- Drill floor control system
- Power management system
- Thruster control system
- Dynamic positioning system
- Integrated automation system
- Emergency shutdown system
- Ballast control system

- High configurability
- 10,000's of I/O
- 1000's of functions

- Many vendors
- OS, Firmware, Applications, Network, ++

2014 © Marine Cybernetics | www.marinecyb.com
MODERN WHEEL HOUSE
WHY DO SOFTWARE ERRORS OCCUR?

- **SW is written by humans**, and humans make errors…
- Most control system software deliveries are unique and **tailor-made** (configured) for the vessel
- A lot of SW functions and configuration is written under a **tight time schedule** and **finished late** in the projects
- Control systems are complex with **many signals and interfaces**
- **Integration of systems** from different vendors is challenging and error-prone
- Frequent **updates and patches** introduce new problems
- Difficult to account for all operational scenarios, failures and **situations that may occur** at the design stage
WHAT IS HIL TESTING?
HIL TESTING – VIRTUAL TEST ENVIRONMENT

Normal operation

Cruise controller I/O

- Commanded throttle
- Speed measurement

HIL testing

Cruise controller I/O

- Commanded throttle
- Simulated speed measurement
HIL TESTING – VIRTUAL TEST ENVIRONMENT

Normal operation

DP system

I/O

Commands

Measurements

HIL testing

DP system

I/O

Commands

Simulated measurements
Types of testing:

1. **Functional testing** to see if the target system software and hardware work as specified.

2. **Failure testing** to check if the control system software and hardware is sufficiently resilient to relevant failure situations.

3. **Performance testing** to see if the control system is tuned to perform with sufficient accuracy – if requested.
COMPANY HISTORY

Identifying software issues:

- Reducing risk for incidents and accidents
- Reducing risk for off-hire and non-productive time
- Securing safe and reliable operations
- Securing flawless startups

Safe software – safe operation
HOW TO CREATE A HIL TEST APPLICATION?
CHOSE AN EXCELLENT MODELING PLATFORM

- Marine Cybernetics has for 13 years performed all modeling in Simulink. We are currently using M2013b.

- Marine Cybernetics has about 40 employees that daily use the following Mathworks products:
  - MATLAB
  - Simulink
  - Simulink coder
  - Embedded Coder

- We have been self-going, but have very good experience with the support team at Mathworks.
# HIL Simulation Models Created with Simulink

<table>
<thead>
<tr>
<th>Hydrodynamics</th>
<th>Propulsion</th>
<th>External loads</th>
<th>Power system</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 6DOF vessel motion</td>
<td>• Azimuths, pods, tunnels,</td>
<td>• Mooring</td>
<td>• Power generation/distribution</td>
</tr>
<tr>
<td>• Wave, wind current</td>
<td>main props, rudders,</td>
<td>• Hawsers</td>
<td>• Load limitation</td>
</tr>
<tr>
<td>loads</td>
<td>...</td>
<td>• Pipe laying</td>
<td>• Buses and breakers</td>
</tr>
<tr>
<td></td>
<td>• Thrust losses and</td>
<td>• Ploughing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>interactions</td>
<td>• Risers</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Streamer</td>
<td></td>
</tr>
<tr>
<td>Drilling system</td>
<td>Sensors</td>
<td>Environ-</td>
<td>Failure modes</td>
</tr>
<tr>
<td>• Draw work</td>
<td>• DGPS</td>
<td>ment</td>
<td>• Inherent electro/mechanical</td>
</tr>
<tr>
<td>• Top drive</td>
<td>• HPR</td>
<td></td>
<td>• Signal failures</td>
</tr>
<tr>
<td>• Pipe racker</td>
<td>• Artemis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Roughneck</td>
<td>• Gyros</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Heave</td>
<td>• VRU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>comp.</td>
<td>• Wind</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Wind sensors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Main benefits of modeling in Simulink:**

- Quickly evaluate your model and validate the simulation results.
- Vast library of standard Simulink blocks available out the box.
- Low level programming skills not needed.
- Self-documenting if used right.
- Code generation rocks!
AUTOMATIC CODE GENERATION WITH EMBEDDED CODER

C++ code is generated from Simulink models by the Embedded Coder.

A PC 32-bit target platform is used (Windows).

Embedded Coder can generate a C++ class interface to model code.

Marine Cybernetics adds a Java Native Interface wrapper to the generated C++ code.

- Control over access to model data
- Ability to multiply instantiate model classes
- Easy integration of model code into C++ programming environments
BRIDGING SIMULINK AND JAVA

- The JNI code implements a defined Java Interface so that any Simulink model will structurally look the same for the calling Java application.
- Example of interface methods:
  - initialize
  - step
  - terminate
  - setParameter
  - setInputSignal
  - getOutputSignal
- Simulink and code generation is a powerful tool for generating models that can be incorporate in a generic software environment.
- For Marine Cybernetics it has been a great success to be able to combine a strong modelling team with dedicated software developers. This has been possible due to:
  - **Automatic code generation**
  - **Clear system boundaries and responsibilities**
CYBERSEA HIL SIMULATOR PLATFORM

- CyberSea is designed to bridge the gap between physical modeling and application development.
- Automates fundamental parts in the process of building full-blown HIL test applications from simulation models.
- Includes generic GUI functionality and configuration tools.
- Includes important HIL features such as built-in signal failure modes, logging, trending, scenario management and web interface.
- Includes an extensive library of I/O interfaces for communication between HIL simulators and control systems.
- CyberSea includes a generic framework for desktop applications. It provides functionality for saving state, connecting actions to menu items, toolbar items and keyboard shortcuts, and window management.
3D graphics is a powerful tool when performing HIL testing of drillfloor systems

However;
The most important is the simulator Simulink models running in the background and the HIL framework
ANTI-COLLISION / ZONE MANAGEMENT SYSTEM: EXAMPLES OF TARGETED FUNCTIONS

- Monitoring
- Alarm and messaging functionality
- Prevent machine from entering other machines zone
- Stop machine if other machine is entering the zone
- Anti-collision release
- Anti-collision ignore
- Stop of machines on non-healthy position data
- Compliance to documentation
EXAMPLES OF TEST FACILITIES
REMTELY LOCATED PC’S FOR MACHINE SIMULATION AND 3D VISUALIZATION
# CYBERSEA SIGNATURE

## Tests

<table>
<thead>
<tr>
<th>Test Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CyScan 1 - Slowly drifting range measurement</td>
<td>-00:57</td>
</tr>
<tr>
<td><strong>Test description</strong></td>
<td></td>
</tr>
<tr>
<td>✅ Validation - Station-keeping performance in sea state SMOOTH</td>
<td>01:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes in sea state SMOOTH</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Station-keeping performance in sea state MODERATE with blackout on port side</td>
<td>01:00</td>
</tr>
<tr>
<td>✅ Validation - Station-keeping performance in sea state MODERATE with blackout on starboard side</td>
<td>01:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes in sea state MODERATE with blackout on port side</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes in sea state MODERATE with blackout on starboard side</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes with loss of all group A sensors and pos-refs in sea state MODERATE</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes with loss of all group B sensors and pos-refs in sea state MODERATE</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes with loss of all group C sensors and pos-refs in sea state MODERATE</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ Validation - Station-keeping performance in sea state ROUGH</td>
<td>01:00</td>
</tr>
<tr>
<td>✅ Validation - Setpoint changes in sea state ROUGH</td>
<td>03:00</td>
</tr>
<tr>
<td>✅ CyScan 1 - Increased noise in position measurement (range and bearing)</td>
<td></td>
</tr>
<tr>
<td>✅ CyScan 1 - Slowly drifting range measurement</td>
<td></td>
</tr>
<tr>
<td>✅ DGPS 1 - Increased noise in both east and north position from DGPS</td>
<td>02:00</td>
</tr>
</tbody>
</table>

Remaining: 80 minutes, 52 tests
<table>
<thead>
<tr>
<th>New Buildings/Retrofits</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 Platform Supply Vessels (PSV)</td>
<td></td>
</tr>
<tr>
<td>15 Anchor Handling Tug Supply (AHTS)</td>
<td></td>
</tr>
<tr>
<td>6 Emergency Rescue Recovery Vessels (ERRV)</td>
<td></td>
</tr>
<tr>
<td>23 Offshore Construction Vessels</td>
<td></td>
</tr>
<tr>
<td>• ROV, Diving, IMR, Well intervention</td>
<td></td>
</tr>
<tr>
<td>56 Drilling units (semi/drillship/jack-up)</td>
<td></td>
</tr>
<tr>
<td>1 Seismic Vessel</td>
<td></td>
</tr>
<tr>
<td>12 Shuttle Tankers</td>
<td></td>
</tr>
</tbody>
</table>
CONCLUSION

HIL testing is an excellent verification method for control systems

Choose the right tools for the different jobs