

# Using Soar with Simulink for Pilot Decision Making

SOAR Workshop 2024

---

Paul Schuricht

29 May 2024

**BLUEBEAR**   
A Saab Company

# Overview

---

- Background
- Pilot Modelling
- Why Soar?
- What is Simulink?
- Linking SOAR and Simulink
- Next Steps
- Issues Encountered
- Summary

# Background

---

- BlueBear
  - Autonomy and UAV Company that also does modelling and simulation
  - Started in 1999
  - UK based
    - Bedford, ~1 hour north of London
  - Acquired by Saab in Aug 2023
  - <https://bbsr.co.uk/>
- Paul Schuricht
  - University of Southampton
    - BSc (Hons) Aero & Astro
    - PhD
  - USAF Academy
    - Post-Doc Research (2 Years)
  - Lockheed Martin UK (10 Years)
  - BlueBear (12 Years so far...)
- Aerospace, Modelling and Simulation – not Computer Science!

# Pilot Modelling

---

- Separate out the decision making from the flying
  - Decision Making
    - Provides the aim/goal
    - What heading should fly?
    - Should I turn towards, or away from, another aircraft?
    - Mission importance or my survival?
  - Flying
    - Using the aircraft controls to achieve the aim/goal
- Have an existing pilot model in Simulink
  - FSM (Stateflow) for Decision making
  - Simulink for Flying
- Looking to replace FSM with Soar
- Pilot Model Uses
  - Current
    - Experimentation in virtual environment
  - Future
    - Synthetic Training
    - Loyal Wingman / CCA

# Why SOAR

---

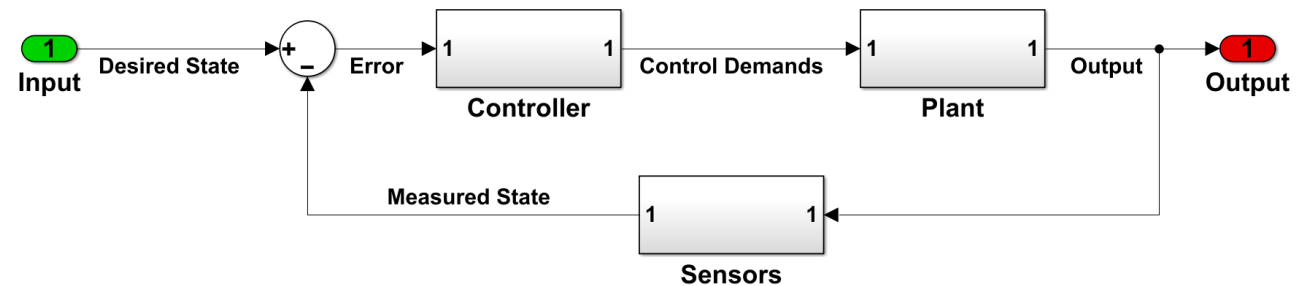
- Colleague attended SoarTech Presentation at I/ITSEC 2022
  - “Introduction to Cognitive Systems for Modeling and Simulation”, Jones & Schmorrow
- This prompted research into Cognitive Systems
- Selected Soar as candidate for our pilot models
  - Soar Tutorial provides good introduction
  - TACAIR-SOAR shows something similar has been accomplished before
- Worked through Soar tutorial
- TankSoar provides a great example
  - Multiple sensors with different capabilities
  - Responding to other entities

# What is Simulink?

---

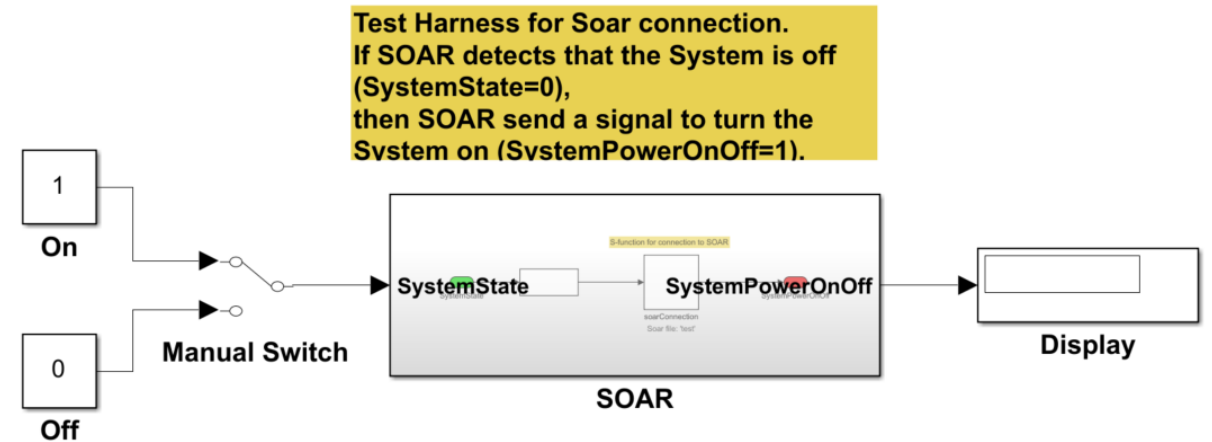
- MATLAB based block diagram environment used
- Widely used in Automotive and Aerospace
- Used to design/model systems and simulate them
- Within BlueBear:
  - Flight/Vehicle Dynamics
  - Control Systems
- Can be autocoded to C/C++ and DLL
  - Used to provide UAV/aircraft model DLLs to run in real-time simulations
- Can incorporate external C/C++ code into Simulink models (S-Function)

- BlueBear have a long history of using Simulink
  - Library of models and subsystems available for reuse



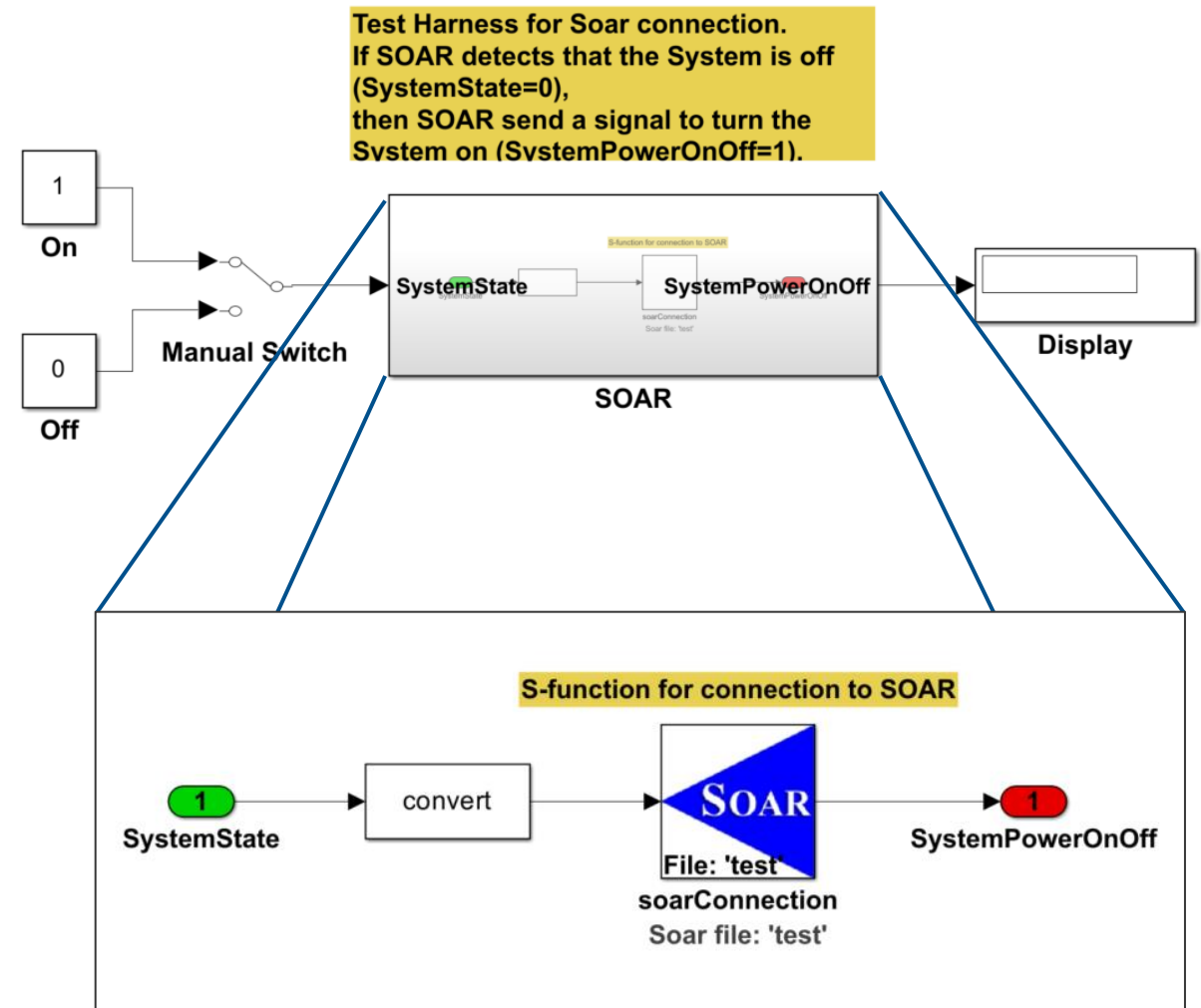
# Linking SOAR and Simulink (Simulink)

- Created an S-Function to link Soar to Simulink
  - S-Function uses SML to interface with Soar
- Simple Test Model
  - While SystemState is 1, Soar does nothing (SystemPowerOnOff 0)
  - Manually set the SystemState to 0, Soar tries to turn it back on (SystemPowerOnOff 1)
- Can set the .soar file to load from Simulink
- Can set the Agent Name from Simulink
  - Useful if you want multiple Agents



# Linking SOAR and Simulink (Simulink)

- Created an S-Function to link Soar to Simulink
  - S-Function uses SML to interface with Soar
- Simple Test Model
  - While SystemState is 1, Soar does nothing (SystemPowerOnOff 0)
  - Manually set the SystemState to 0, Soar tries to turn it back on (SystemPowerOnOff 1)
- Can set the .soar file to load from Simulink
- Can set the Agent Name from Simulink
  - Useful if you want multiple Agents





# Linking Soar and Simulink (Soar)

---

- Input-link (from Simulink)
  - System.Status (**on** or **off**)
- Output-link (to Simulink)
  - System.Power (**on** or **off**)
- S-function converts from **on/off** in Soar to **0/1** in Simulink

```
1 # If the System is off propose turning it on.
2 sp {propose*system*on
3     (state <s> ^type state
4         ^io.input-link <il>)
5     (<il> ^System.Status off)
6 -->
7     (<s> ^operator <o> +)
8     (<o> ^name system-on
9         ^action <a>)
10    (<a> ^System.Power on)}
11
12 # If System on is proposed then apply it
13 sp {apply*system*on
14     (state <s> ^type state
15         ^io.output-link <ol>
16         ^operator <o>)
17     (<o> ^name system-on)
18 -->
19     (<ol> ^System.Power on)}
20
21 # Remove a completed command
22 # If an attribute on the output link has status complete remove it
23 sp {remove*complete*commands
24     (state <s> ^operator <o>
25         ^io.output-link <ol>)
26     (<ol> ^<attr> <val>)
27     (<val> ^status complete)
28 -->
29     (<ol> ^<attr> <val> -)}
```

# Next Steps

---

- Ideas for Extending the Simulink/Soar Simulation
  - 2 aircraft approaching each other
    - One (or both) with Soar “pilot”
  - Soar pilot initiates manoeuvre to avoid collision when aircraft get too close
  - Add waypoints to fly to
    - Avoid collisions while continuing with waypoint following
  - Soar pilot monitors fuel state
    - “Head home” when fuel gets low
  - Communication between pilots
    - Provide state updates to each other
    - Lead provides orders to Wingman
      - Collaborative behaviours

# Issues Encountered

---

- Some issues with developing more complex simulations
  - Early days
  - Likely to be User Error!
  - Hope to pick up some useful knowledge from the workshop
- Haven't managed to link Soar running in Simulink to the Soar Debugger
- Working Memory is filling up quickly
  - Slows simulation
  - Tried DestroyWME()
- Some issues getting proposed actions to trigger

- Debugging Soar
  - Visualise the Working Memory
    - See outputSoarGraph() below
  - Can end up with a lot of files if done every timestep!

```
static void outputSoarGraph(sml::Kernel* pKernel, sml::Agent* pAgent, bool output)
{
    if (output)
    {
        PLOG_DEBUG << "Generating graph output";
        // Soar command line command to output a graph
        std::string cmd = "visualize architectural-wmes on" ;
        std::string cmd2 = "visualize wm" ;|
        // Execute the command
        char const* pResult = pKernel->ExecuteCommandLine(cmd.c_str(),pAgent->GetAgentName()) ;
        char const* pResult2 = pKernel->ExecuteCommandLine(cmd2.c_str(),pAgent->GetAgentName()) ;
    }
}
```

# Summary

---

- Demonstrated that Soar can be linked with Simulink with a simple example
  - This needs to be developed to more complex examples for practical use
  
  - Thank you for letting me present
  - Any advice, feedback or questions welcome
- Contact Details:
  - Paul Schuricht
  - [paul.schuricht@bluebearsystems.co.uk](mailto:paul.schuricht@bluebearsystems.co.uk)
  
  - Google Groups require a gmail address
  - [paul20bluebear@gmail.com](mailto:paul20bluebear@gmail.com)
    - Soar Cognitive Architecture Research Community
    - Soar Cognitive Architecture Help

# Questions?

---

**BLUEBEAR** 

A Saab Company