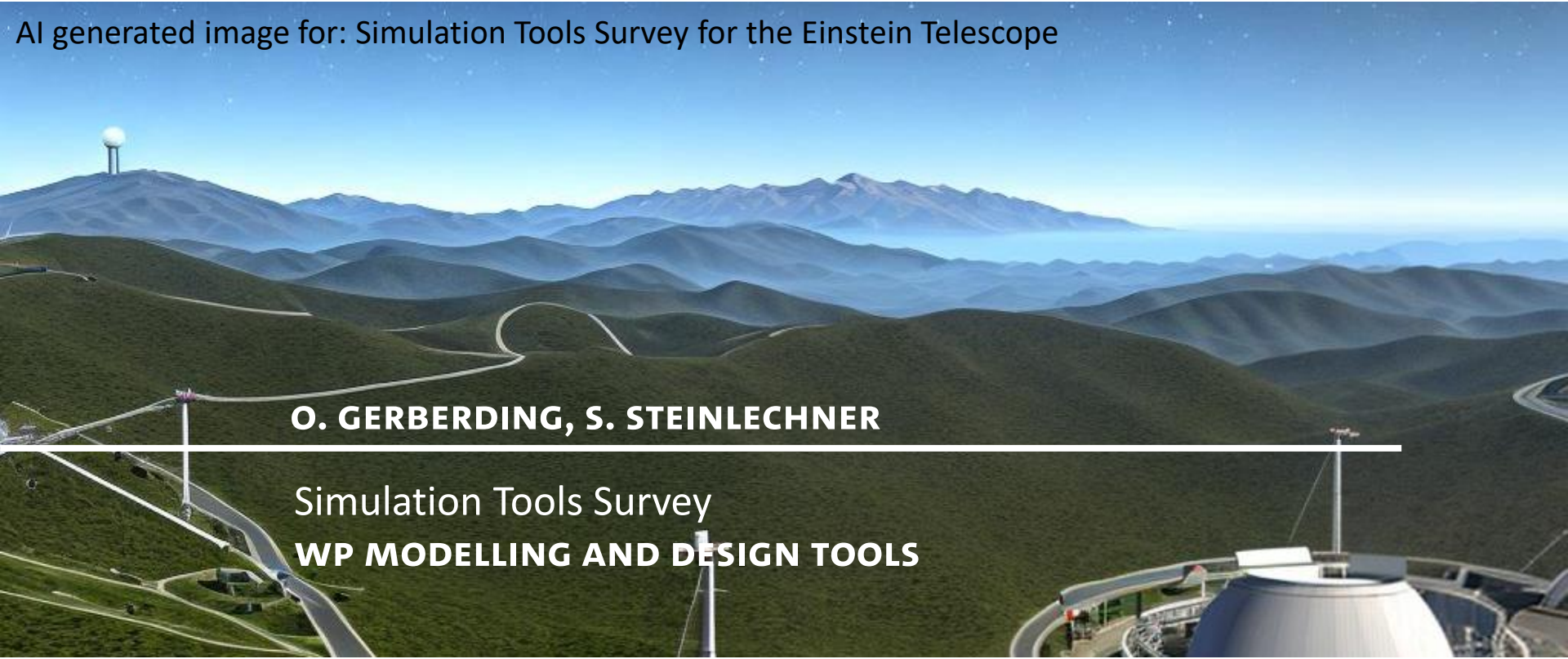
AI generated image for: Simulation Tools Survey for the Einstein Telescope



O. GERBERDING, S. STEINLECHNER

Simulation Tools Survey

WP MODELLING AND DESIGN TOOLS

SIMULATION TOOL SURVEY

Goal:
A list of all software tools that are, and might be, used to model and simulate ET or parts of ET (ET instrument)

Approach:

E-Mail to the ISB (January 27th 2023)

Results:

31 software tools (commercial, open source and everything in-between)

https://wiki.et-gw.eu/ISB/Interferometer/ModelDesignTools/Swlist

SIMULATION TOOL SURVEY RESULTS – IFO 1

Name	Application (what is it used for in ET/ISB)	Type (open source, commercial, closed source, unclear)	Programming Languages	Maintainer	Status	Link
Finesse 3	Interferometer simulation	open source	python	A. Freise et al.	active	https://git.ligo.org/finesse/finesse3
Lightsaber	Nonlinear, time-domain simulation of interferometers and their control system	open source	python	J Harms	active	https://gitlab.et-gw.eu/et/isb/active-noise-mitigation/controls-simulations/-/tree/main/Lightsaber_X
IfoCAD	3D ray tracing, optimisation	to be open sourced	C++	G. Wanner et al	active	
OptoCAD	2D Gaussian ray tracing	unclear	Fortran	R. Schilling†	abandoned	https://git.ligo.org/IFOsim/optocad
Opticle	interferometer simulation	open source	Matlab-based	M. Evans	abandoned	https://githuhttps://github.com/nicolassmith/lentickleb.com/Optickle/Optickle
Optickle2	interferometer simulation	open source	Matlab-based	M. Evans	abandoned?	https://github.com/Optickle/Optickle/tree/Optickle2
Lentickle	interferometer control simulation	open source	Matlab-based	N. Smith-Lefebvre	abandoned?	
FogPrime13	FFT interferometer simulation	unclear	unclear	H. Yamamoto	unclear	
OSCAR	Steady state optical interferometer simulation	open source	matlab	J. Degallaix	active	https://github.com/Jerome-LMA/oscar

SIMULATION TOOL SURVEY RESULTS – IFO 2

Name	Application (what is it used for in ET/ISB)	Type (open source, commercial, closed source, unclear)	Programming Languages	Maintainer	Status	Link
MIST	Modal Interferometer Simulation Toolbox	open source	Matlab-based	G. Vajente	abandoned?	https://sourceforge.net/projects/optics-mist/
SIS	Stationary Interferometer Simulation	unclear	Matlab-based	H. Yamamoto	unclear	https://labcit.ligo.caltech.edu/~hiro/SIS/
Zemax	Ray tracing, optics simulation	commercial				https://www.zemax.com/
A La Mode	mode matching tool	open source	Matlab-based	N. Smith-Lefebvre	abandoned?	https://github.com/nicolassmith/alm
gtrace	ray tracing and CAD python code	open source	python	Y. Aso	unclear	https://github.com/asoy01/gtrace
e2e	time domain simulation framework of opto-mechanical system	unclear	C++ + GUI in JAVA	H.Yamamoto	abondoned	https://labcit.ligo.caltech.edu/~e2e/
JamMT	mode matching	unclear	Java	N. Lastzka	abandoned	https://git.ligo.org/IFOsim/jammt
FRED Optical Engineering Software	3D raytracing, stray light	commercial (free license for PhD students)	GUI + basic scripting	Photon Engineering, LLC	active	https://photonengr.com/fred-software/

SIMULATION TOOL SURVEY RESULTS – NOISE BUDGET

Name	Application (what is it used for in ET/ISB)	Type (open source, commercial, closed source, unclear)	Programming Languages	Maintainer	Status	Link
pygwinc	Interferometer Noise Calculator	open source	python	C. Wipf, J. Rollins et al.	active	https://git.ligo.org/gwinc/pygwinc
Noise Budget Simulink Tool	noise budget tool	unclear	Matlab/Simulink-based	C. Wipf	unclear	https://svn.ligo.caltech.edu/svn/aligonoisebudget/trunk/

SIMULATION TOOL SURVEY RESULTS — SYSTEMS, CONTROL, SUSPENSIONS, SIGNAL PROCESSING

Name	Application (what is it used for in ET/ISB)	Type (open source, commercial, closed source, unclear)	Programming Languages	Maintainer	Status	Link
Matlab Simulink	linear system and control modeling	commercial	Matlab		active	https://www.mathworks.com/products/matlab.html
Spicypy	designing, optimizing control loops, and signal processing	open source (Apache License v 2.0)	python	A. Basalaev, C. Darsow-Fromm, O.	active	https://gitlab.com/pyda-group/spicypy https://pypi.org/project/spicypy/
GWpy	General signal processing, (mock) GW data analysis, simple access to LIGO time series data	open source (GNU GPL v3.0)	python	D. Macleod et al	active	https://gwpy.github.io/
python-control	Modeling control loops (includes limited MIMO, nonlinear systems support)	open source (BSD-3-Clause)	python		active	http://python-control.org/
Matlab LTPDA	object-oriented approach to data analysis and system modeling (LISA Pathfinder)	free	Matlab	M. Hewitson	stale?	https://www.elisascience.org/ltpda/index.html
Octopyus	Simulating seismic isolation systems for 3rd gen GWD	unclear	python	M. Razzano et al	active	

SIMULATION TOOL SURVEY RESULTS – ELECTRONICS & FEA/CAD

Name	Application (what is it used for in ET/ISB)	Type (open source, commercial, closed source, unclear)	Programming Languages	Maintainer	Status	Link
SPICE (LTspice)	circuit simulator software	free	Analog Devices		active	www.analog.com/
LISO	freq-domain electronics simulation	unclear	C	G. Heinzel	stale?	
Zero	python-reimplementation of LISO	open source	python	S. Leavey	stale?	https://github.com/SeanDS/zero/
Comsol	FEA	commercial				https://www.comsol.de/
Ansys	FEA	commercial				https://www.ansys.com/
SolidWorks	3D modeling of mechanical parts	commercial				

SIMULATION TOOL SURVEY - CONCLUSION

- Many tools are out there
- Often their status is not well defined
  -> we should start to rely on tools with active maintenance and, if possible, open source tools, for instrument studies
- Certainly not all necessary tools have been covered!
- Please send more relevant tools and any corrections/additions to:
  oliver.gerberding@physik.uni-hamburg.de
  s.steinlechner@nikhef.nl

WP MODELLING AND DESIGN TOOLS – GENERAL COMMENTS

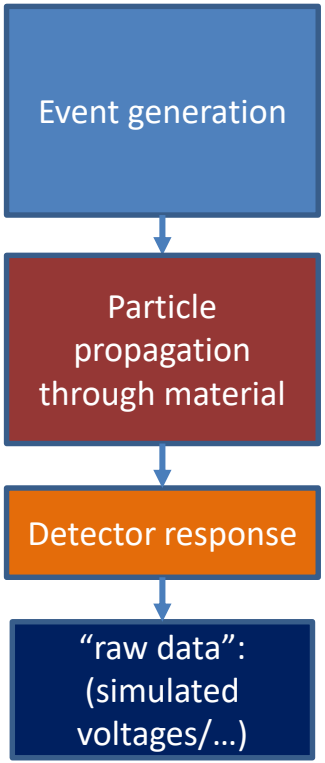
- A lot of software will be used and needed
- Aim of our WP is to work towards
    - Best practices for SW development (if done by/with/for ET-ISB)
    - Good and efficient software documentation recommendations
    - Definition of software interfaces and data formats
    - Regularly available SW training
    - Established review and management procedures (especially for internally dev. SW)
- Each analysis done in software needs to be reproducible, this requires
    - defined and available set of input/design parameters/files
    - defined and available software and library versions
    - available analysis code
    - each of the above might become its own repository

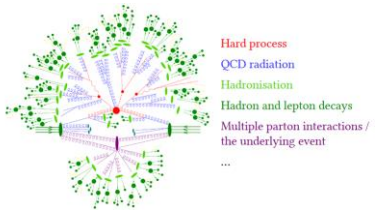
FULL DETECTOR SIMULATION AKA “DIGITAL TWIN” AN EXAMPLE FROM ATLAS COLLABORATION

ARTEM BASALAEV

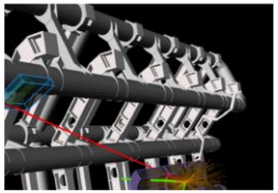
ATLAS detector simulation – very brief

- *Disclaimer: I only worked on this as a PhD student and by no means an expert*

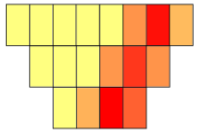




- Independent of detector materials and conditions
- Done with **external, well-established software**

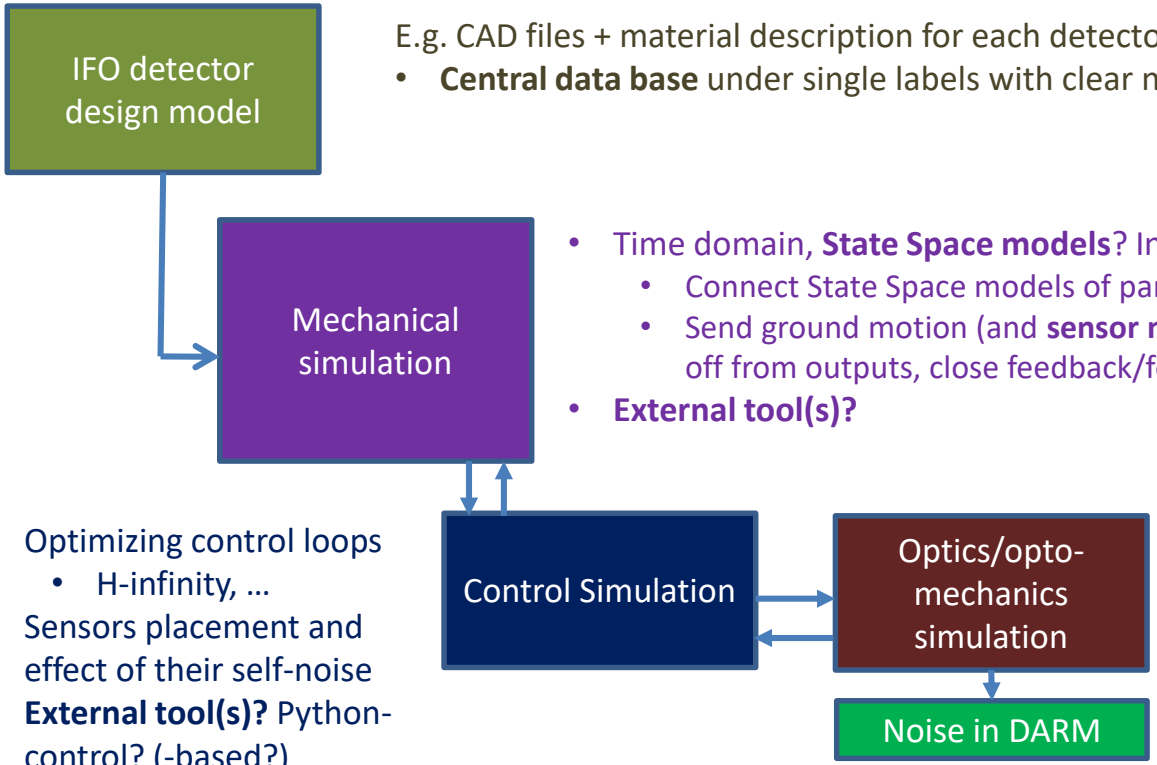


- Dependent on detector materials/layout
- Create **detector geometry models in-house**, which are **stored in central data base**
- Simulation itself by **external, well-established software**



- Software packages for each subdetector **produced fully “in-house”** by different subgroups
- **Centrally managed**: packages integrated into ATLAS simulation software chain, with central gitlab repo and strict code review/control

Can we do something similar for IFOs/ET? Maybe?...



E.g. CAD files + material description for each detector part, forms full "IFO geometry"

- **Central data base** under single labels with clear naming convention?

- Time domain, **State Space models**? In that case, modular design:
  - Connect State Space models of parts to each other to form bigger parts
  - Send ground motion (and **sensor noise**) into inputs and its propagation read off from outputs, close feedback/forward loops with extra connections
- **External tool(s)?**

- Optimizing control loops
  - H-infinity, …
- Sensors placement and effect of their self-noise
- **External tool(s)?** Python-control? (-based?)

- Interplay of cavities and radiation pressure
- **External tool(s)?** Finesse 3?