Hopes and Dreams: One Transfer Function Fitting Program to Rule Them All

J. Kissel

GWADW, May 25th 2016

Context:

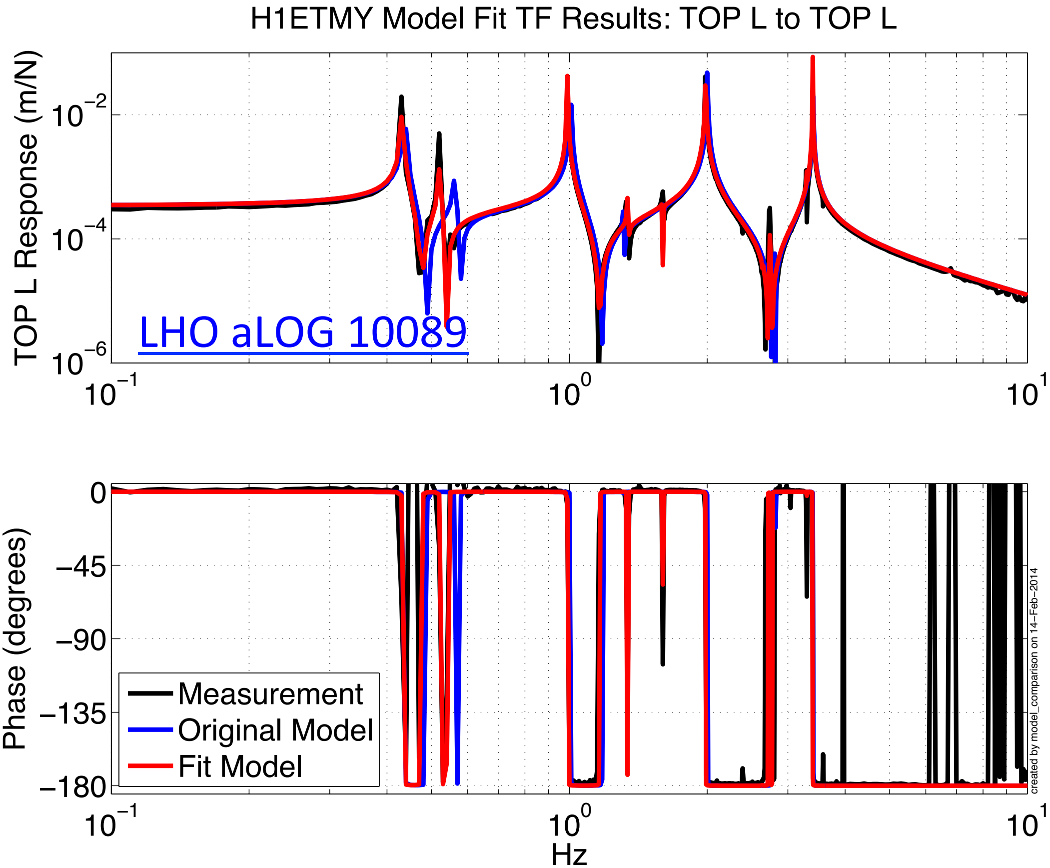
Fitting measured transfer functions

- An integral part of classical control design and commissioning is system identification
- Because it's "easy" many people have created functions for their specific need
- Because it's "hard" many academics have cooked up several algorithms
- A variety of end-use cases scenarios
  - To make filters which cancel out real-world details to make control system behave like idealized system
  - To help design stable-and-robust control filters around plant
  - To estimate underlying physical parameters of plant
  - To predict other immeasurable transfer functions of plant
  - To aide in noise budget modeling of sensor signals
- Need will be ever-present
- What exists that gets best results: typically written by physicists (non-software engineers) for immediate need / end-goal and poorly maintained after initial use, but used as "hand-me-down" to their students and mentees

When Have I Fit A Transfer Function?

- Seismic (using *N4SID*)
  - Low-Q, MIMO Plant fitting for Noise Modeling
- Suspensions (using *Home-brewed Matlab*)
  - Parameter Estimation for Dynamical Model Refinement
- Calibration (using *pre-packaged Python*)
  - Bayesian parameter estimation for uncertainty and covariance
- Interferometer Control (using *VectFit*)
  - High-Q, Transfer-function Ratios for DOF decoupling
- Calibration – Precision Fitting (using *LISO*)
  - Electronics poles & zeros with uncertainty estimation

Home-brewed Matlab



Goals of task:

- Get a very-accurate MIMO dynamical model for use in noise modeling
- Compact state-space representation where one can request/predict arbitrary or poorly measured transfer functions
- Original model built from *many* physical parameters
- Want to know as-built values for underlying parameters, and uncertainty and covariance in the resulting transfer function estimate

Guass-Newton Method T1000458, T1100163

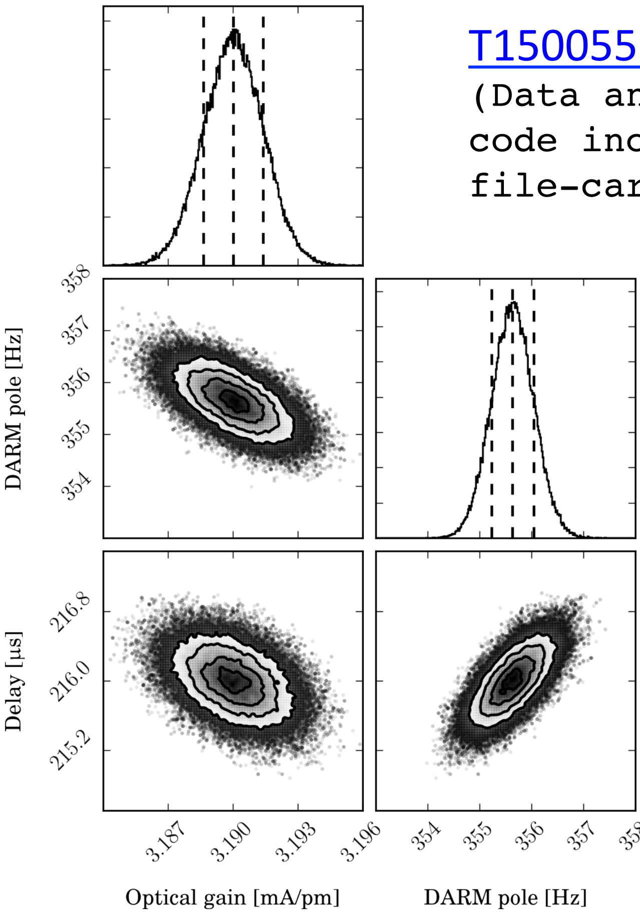
Fisher-matrix based approach predicts *minimum* uncertainty

```
${SusSVN}/sus/trunk/QUAD/H1/QUADTST/SAGM0/Results
2014-09-18_1630_H1SUSQUADTST_M0_DTTTF.mat

${SusSVN}/sus/trunk/QUAD/Common/MatlabTools/QuadModel_Fit/
QuadPend_GaussNewton_fit_v4_X1QUAD06.m
```

Pre-packaged Python

Goals of task:

- Get an accurate and precise estimate of a few parameters, with uncertainty and covariance
- Use posteriors from Bayesian Likelihood analysis to create brute-force estimate of combined transfer functions (instead of traditional analytic propagation of uncertainty)

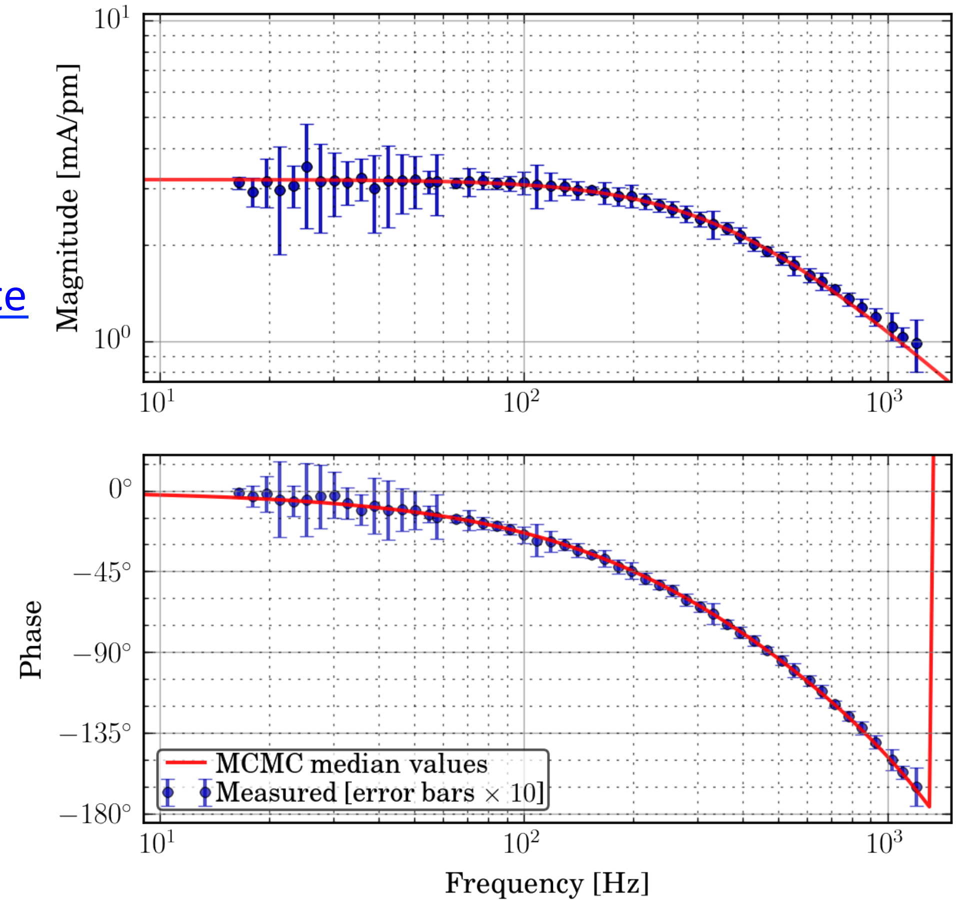


[T1500553](#)

```
(Data and analysis
code included in DCC
file-card)
```

[emcee Website](#)

[emcee Paper](#)



Home-brewed Matlab/Python

From where did I hear about its first successful use?

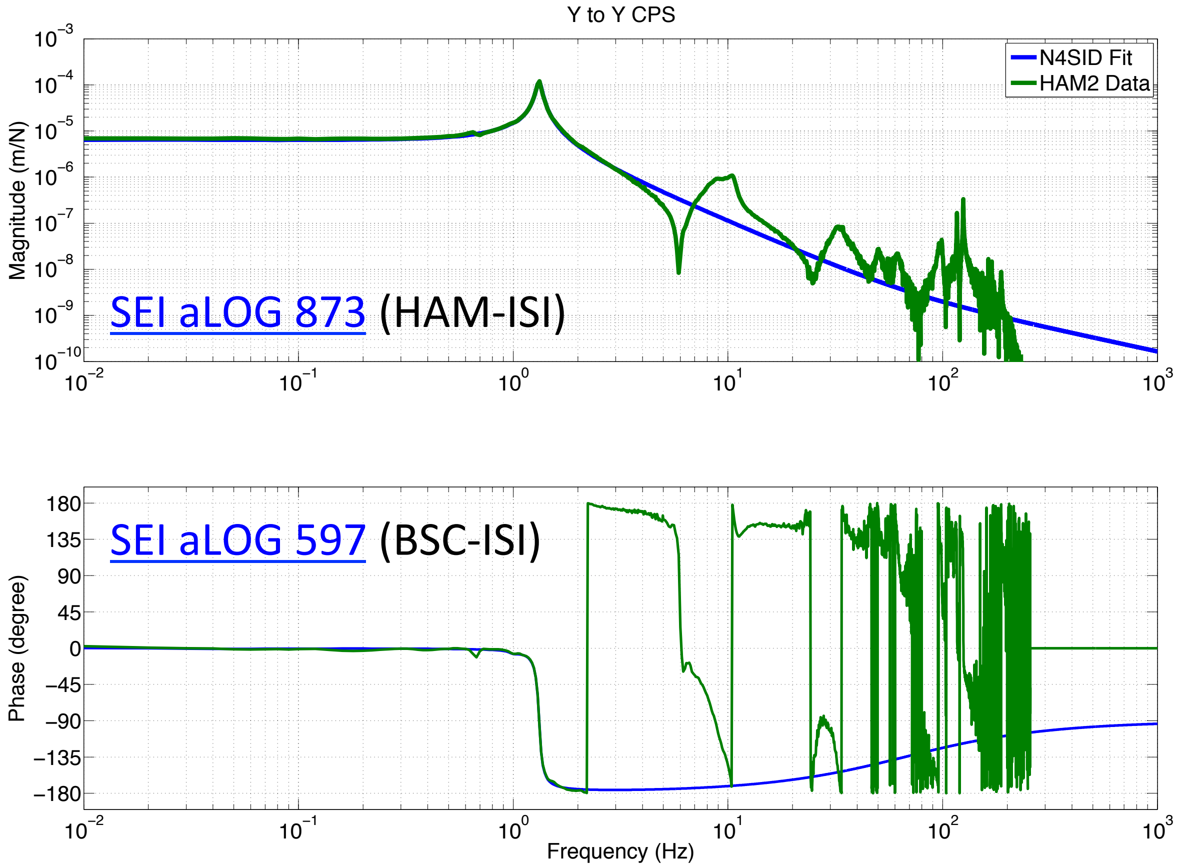
- Fitting the parameters of the As-Built aLIGO QUADs, with Guass-Newton Method, B. Shapiro; T1000458
- Multi-variate likelihood code with (MC)MC e.g. emcee, E. Hall; T1500553
- The whole LIGO GW Astrophysics Parameter Estimation Industry

- Pros
  - Very flexible, lots of smarts can be built in
  - Suits immediate need well
  - Multivariate uncertainty estimation with graphics!
  - (Some) integrated into Matlab already
- Cons
  - Learning curve: easy-to-high
  - "easy" stuff is single-user, single-use code; not well-documented enough to re-use
  - Some fitting routines are "academic" i.e. novel (because the goal is publishing)
  - Takes a while to develop / debug / verify

N4SID: Matlab Built In Tool

Goals of task:

- Get a semi-accurate MIMO dynamical model for use in noise modeling
- Compact state-space representation
- Don't care about under-lying parameters (or their uncertainty)
- Can ignore higher-frequency dynamics



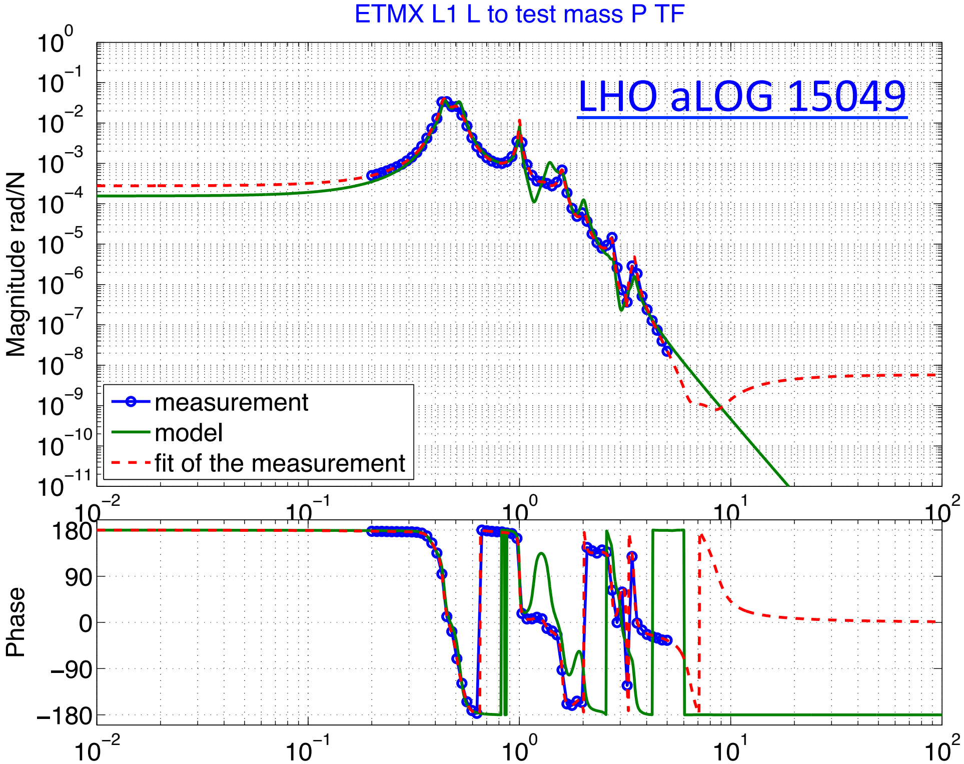

```
SeiSVN/seismic/HAM-ISI/HAM2/Data/Transfer_Functions/Simulations/Undamped/
H1_ISI_HAM2_TF_C2C_Raw_2014_09_16.mat

SeiSVN/seismic/HAM-ISI/Common/HAM_ISI_Model/N4SID_Model/H1HAM2/
HAM2_N4SID_fits.m
```

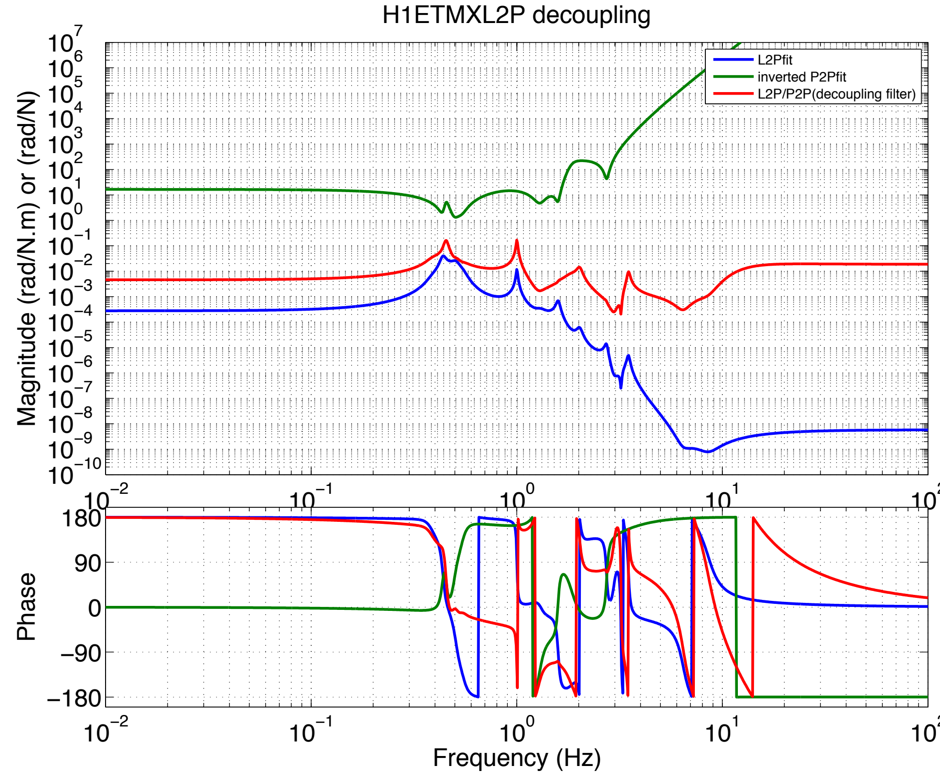
N4SID: Matlab Built In Tool

From where did I hear about its first successful use?
Virgo Super-Attenuator Dynamical Model, V. Boschi; P1000117

- Pros
  - Can ingest both frequency domain and time-domain measurements
  - Good for many-state MIMO systems
  - Built-in Matlab toolbox / suite
  - Some multivariate uncertainty estimation
- Cons
  - Works better with time-domain data, which for low-frequency systems means bigger measurement files
  - Frequency-domain fitting not so good at cross-terms of MIMO data set
  - Tough to weight measurements by coherence
  - learning curve: medium

(happy)VectFit(3, 4)





Goals of task:

- Use measurements to design plant-inversion filters, to reduce frequency-dependent cross-coupling.
- "Measurement" is a ratio of several complicated transfer functions
- Focused solely on region with complicated dynamics, ignore high-frequency poles, right-half-plane poles and zeros
- Desire for least number of poles and zeros, low-impulse response final filter

Vectfit Website (!!)

VectFit3 User Guide (!!)

```
${SusSVN}/sus/trunk/QUAD/H1/ETMY/SAGL1/Data/
2014-11-13_H1SUSETMY_L1_*.txt

${SusSVN}/sus/trunk/QUAD/H1/ETMY/SAGL1/Scripts/
design_H1SUSETMY_L1_L2P_20141113.m
```

(happy)VectFit(3,4)

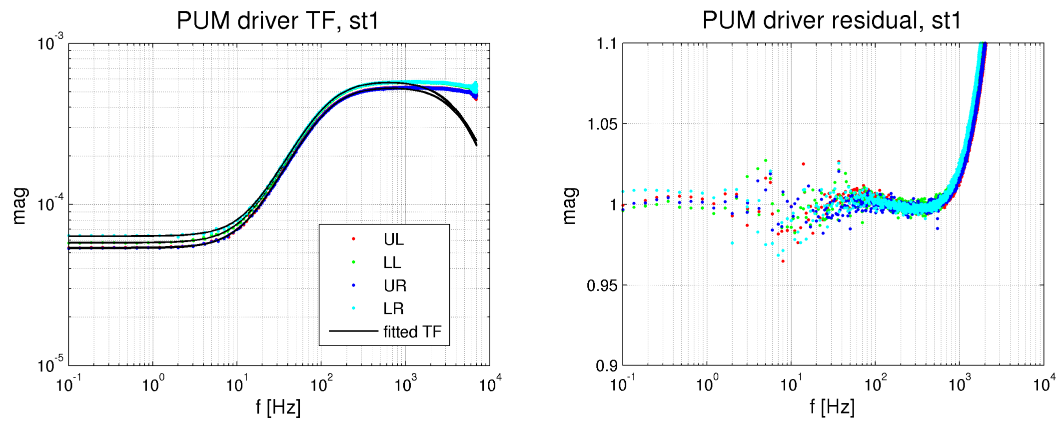
From where did I hear about its first successful use?

Fitting a ratio of transfer functions to create ideal Wiener filter corrections to HEPI sensor correction. J. Driggers, R. Derosa P1000088

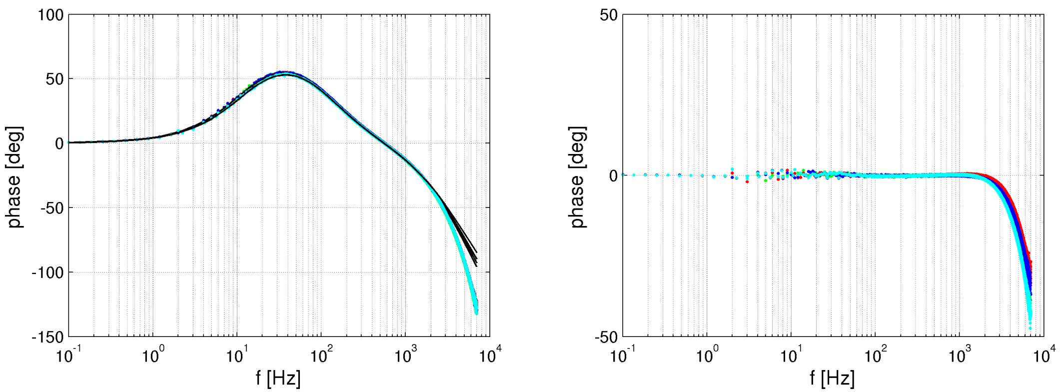
- Pros
  - Integrated into Matlab already

- Cons
  - Finicky results
  - Needs seed poles and zeros
  - Restricted to equal number of poles and zeros (and can only define number of poles)
  - Difficult user-interface means there's lots of copies, improvements, and wrappers.
  - Doesn't drop right-half-plane poles and zeros
  - No uncertainty estimation
  - Multiple versions
  - learning curve: high

LISO



LHO aLOG 21232



Goals of task:

- Very accurate model of poles and zeros
- Poles and zeros used in compensation filters
- Uncertainty estimation on poles and zeros, as well as overall fitted transfer function
- Learning Curve: Medium

LISO Manual

LISO Software Package

```
${CalSVN}/aligocalibration/trunk/Runs/ER8/H1/Measurements/Electronics/
2015-09-01_H1SUSETMX_UIMDriver_State*/*.txt
```

```
${CalSVN}/aligocalibration/trunk/Runs/ER8/H1/Scripts/Electronics/
runFit_H1SUSETMY_PUMDriver.m
```

LISO

From where did I hear about its first successful use?
Fitting of electronics transfer functions for building compensation filters, e.g. Z. Korth; LLO aLOG 5349

- Pros
  - Very rudimentary and robust
  - Some multivariate uncertainty estimation
- Cons
  - text file and gnuplot output / no graphical interface
  - Needs seed poles and zeros
  - Needs integration into Matlab

Goals for TF Fitting Super Suite:

- User-friendly (!!!)

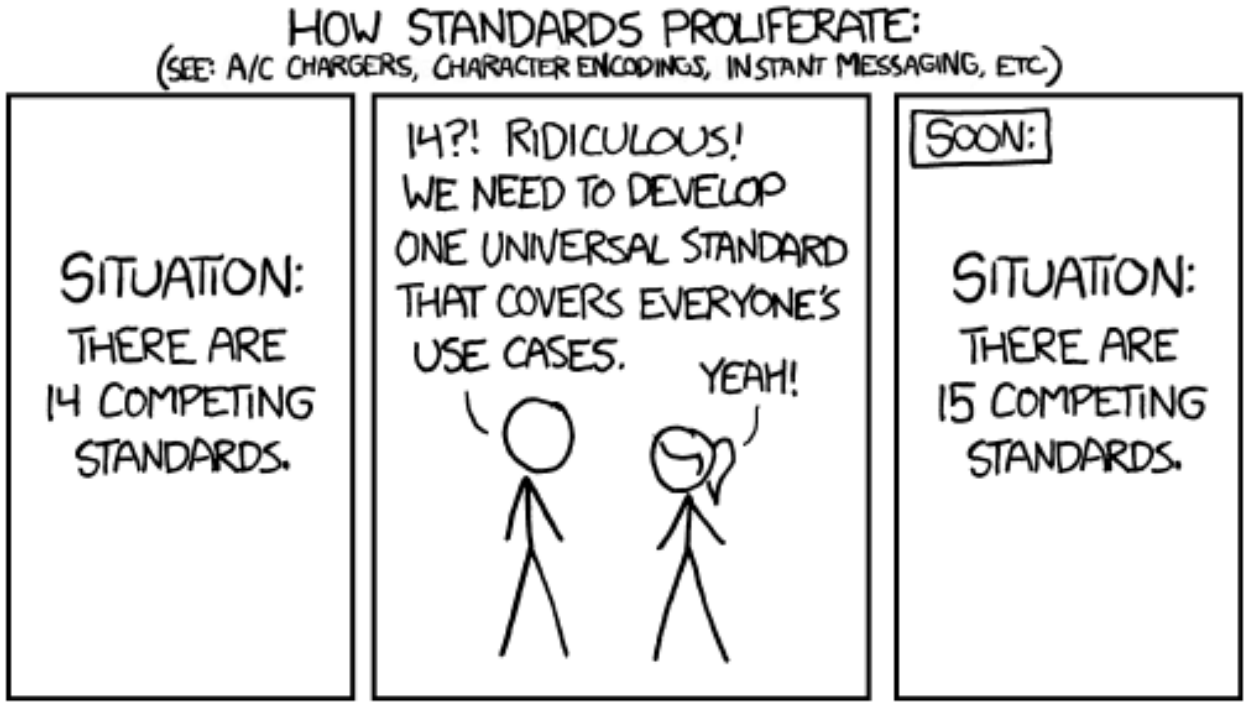
- Output needs
  - choice of poles and zeros, LTI object, or discrete filter
  - uncertainty estimation (including covariance matrix) of output product
  - ability to fit MIMO TFs (using either frequency or time domain data)
  - ability to weight input (coherence, frequency region, emphasis on magnitude or phase)

- More physical and practical restrictions on the output
  - more logic built-in to reduce the fit order
  - limit if not disallow right-half-plane poles and zeros

- Preference to play well with Matlab (a fit result almost always goes immediately into Matlab for use in further control filter design anyways)

- Lightweight
  - fitting turn around time of ~seconds
  - not reliance on cumbersome software like a simulink

Things to Avoid / Classic Software Problems



Credit: xkcd.com

Since all code mentioned to this point is either not or no longer maintained, the above might be impossible, but if we keep it in mind and don't re-invent the wheel, we might turn out OK

"This code is *great*! I'm going to completely re-write it instead of improving it."

Summary

	Matlab Integration	User Friendly	Uncertainty Estimation	Maintained	Inherent Smarts	Flexibility
Homebrew	Yes	Yes (but)	Yes	No	Yes	High
VectFit	Yes	No	No	No	No	Low
LISO	No	Yes (but)	Yes	No	No	Low
N4SID	Yes	No	Yes	Yes (but)	Yes	Low

- There's no one-size fits all transfer function fitting tool on the market
- Most have more cons than pros, but can get the job done with brute force and time
- Un-solveable here… will take time and dedication
- Not-dissimilar from GW Data Analysis parameter estimation
- If we could get the pros of each and create a user-friendly, flexible, well-maintained suite, then it's **Puppies and Rainbows** for everyone!

Whatever happened to SimMechanics?

- Boschi talk at 2014 SUS/SEI Workshop G1401231
- Looks like Nikhef has used it, Caela Barry Paper