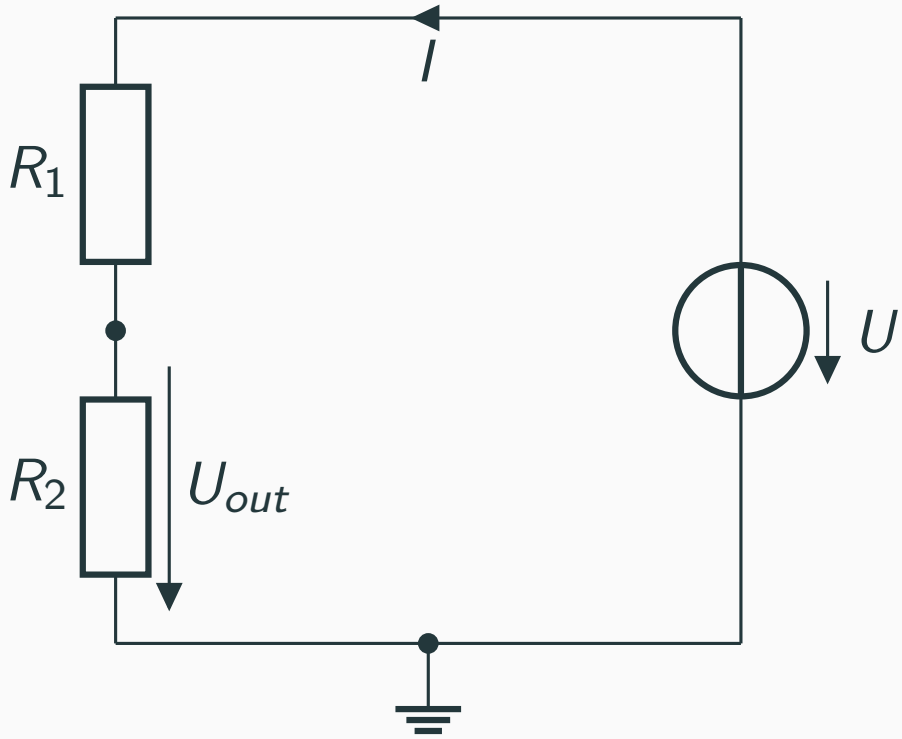
A probabilistic electrical simulator

Jonas Bodingbauer

February 01, 2024

Electrical Circuit: Voltage divider



$$U_{out} = U \frac{R_2}{R_1 + R_2}$$

Idea

- *Components* in a circuit: *expected value* and a *tolerance*.

Idea

- *Components* in a circuit: *expected value* and a *tolerance*.
- *Circuit* gets input $\rightarrow$ response

Idea

- *Components* in a circuit: *expected value* and a *tolerance*.
- *Circuit* gets input $\rightarrow$ response
- The *response* is a random variable, which can be described by a probability distribution.

Idea

- *Components* in a circuit: *expected value* and a *tolerance*.
  $\rightarrow$ **Priors**
- *Circuit* gets input $\rightarrow$ response
- The *response* is a random variable, which can be described by a probability distribution. $\rightarrow$ **Observed**

Idea

- *Components* in a circuit: *expected value* and a *tolerance*. $\rightarrow$ **Priors**
- *Circuit* gets input $\rightarrow$ response
- The *response* is a random variable, which can be described by a probability distribution. $\rightarrow$ **Observed**
- Even more randomness! Noise of devices, Noise of measurement instruments, offsets, ...

Electrical Simulations

- DC Analysis
- AC Analysis (treat everything as a linear component)
- DC Analysis with nonlinearity (e.g. diodes)
- Transient Analysis
- Lots more...

Electrical Simulations

- **DC Analysis**
- **AC Analysis** (treat everything as a linear component)
- **DC Analysis with nonlinearity** DC Analysis with nonlinearity (e.g. diodes)
- Transient Analysis
- Lots more...

Finally, some modelling!

$$\underbrace{\begin{bmatrix} \mathbf{Y}' & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}_{Y} \cdot \underbrace{\begin{bmatrix} \vec{\varphi} \\ \vec{I} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \vec{J} \\ \vec{E} \end{bmatrix}}_{F}$$

Finally, some modelling!

$$\underbrace{\begin{bmatrix} \mathbf{Y}' & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}_{Y} \cdot \underbrace{\begin{bmatrix} \vec{\varphi} \\ \vec{I} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \vec{J} \\ \vec{E} \end{bmatrix}}_{F}$$

AC-Case: Domain changes to complex numbers (Fourier transform)

Finally, some modelling!

$$\underbrace{\begin{bmatrix} \mathbf{Y}' & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}_{Y} \cdot \underbrace{\begin{bmatrix} \vec{\varphi} \\ \vec{I} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \vec{J} \\ \vec{E} \end{bmatrix}}_{F}$$

AC-Case: Domain changes to complex numbers (Fourier transform)

Nonlinear Case: Newton-Raphson Fixpoint iteration

$$\mathbf{Y}(\vec{x_i}) \cdot \vec{x}_{i+1} = -\mathbf{F}(\vec{x_i}) + \mathbf{Y}(\vec{x_i}) \cdot \vec{x_i}$$

Program flow

1. Read SPICE - like file
   voltagediv.cir:

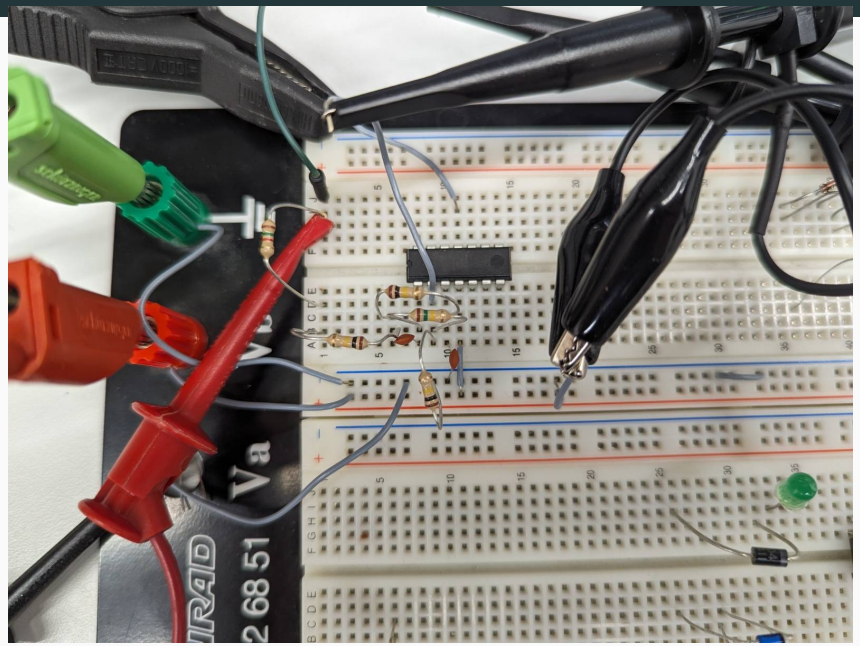
```
V1 0 in DC 2
R3 in 2 11.72
R1 2 3 1.2k
R2 3 0 1.8k
```

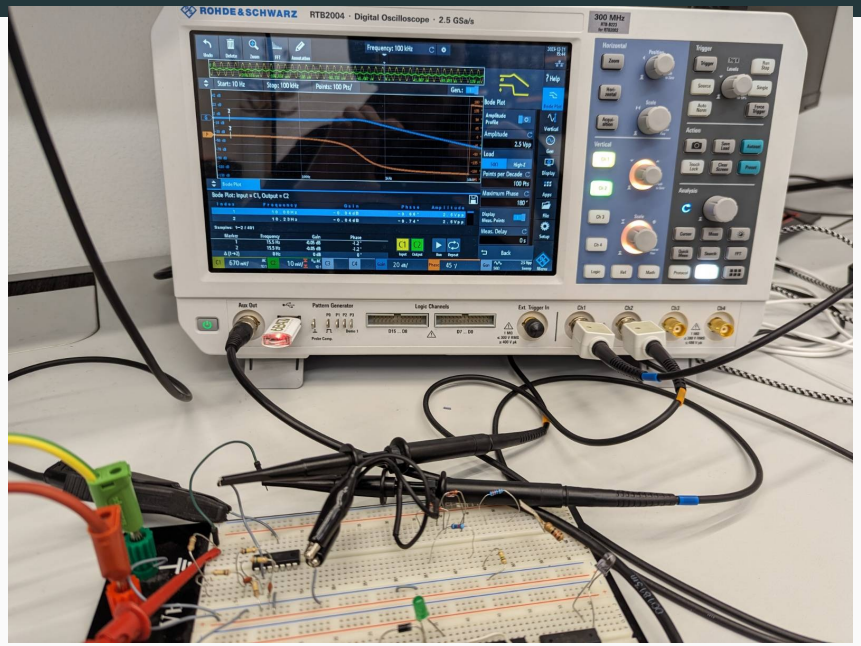
   voltagediv.tol:

```
R3 0.1%
R1 5%
R2 5%
```

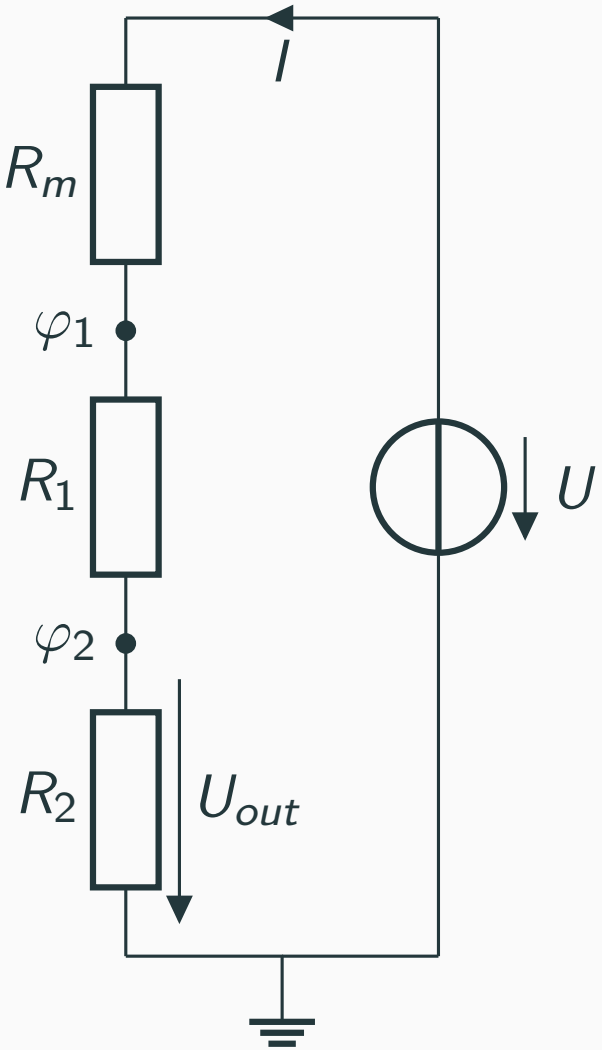
2. Determine the matrix size
3. Read observed data from file(s)
4. Run inference using the simulator as a model

Measurements



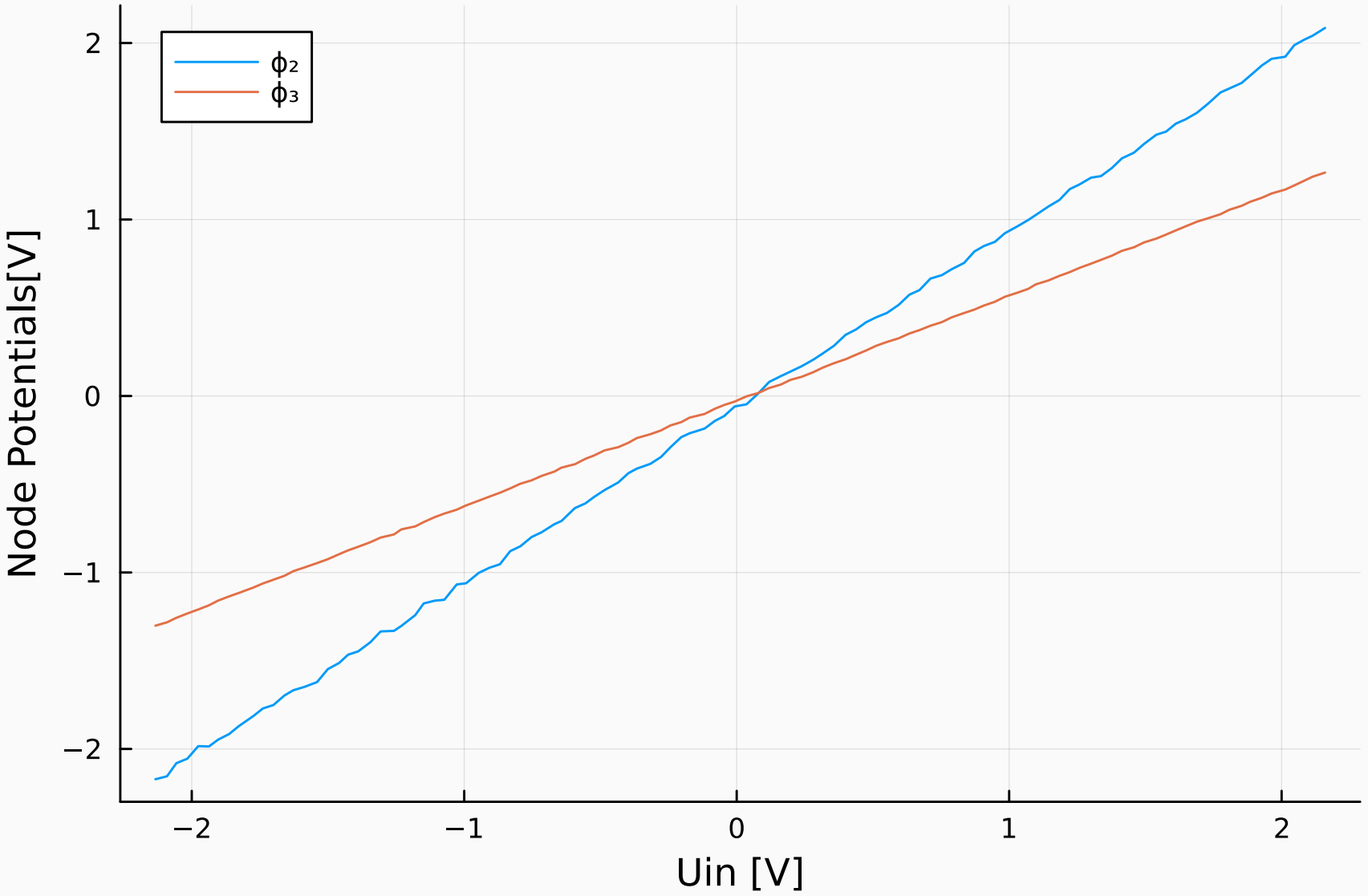


First example extended

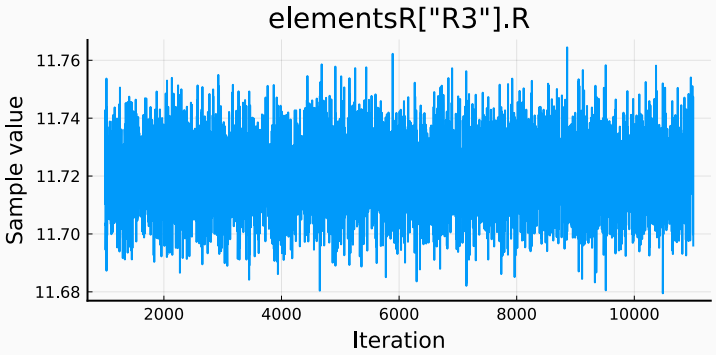


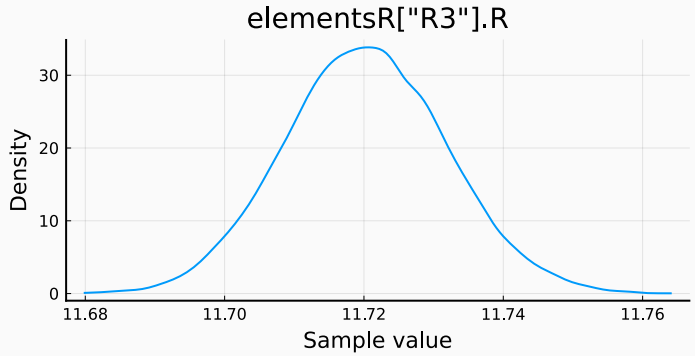
- R_m is an additional (precision) resistor for measuring the current.
- φ_1 and φ_2 are potentials
- Priors:
  - R_1, R_2, normally distributed around nominal value.
  - U_{offset_1}, U_{offset_2} normally distributed around 0.
  - U_{noise_1}, U_{noise_2} uniformly distributed $[0, 0.1]V$.
- Observed: φ_1, φ_2

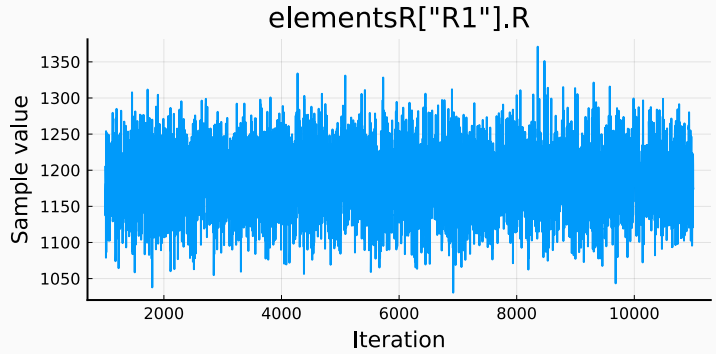
Results

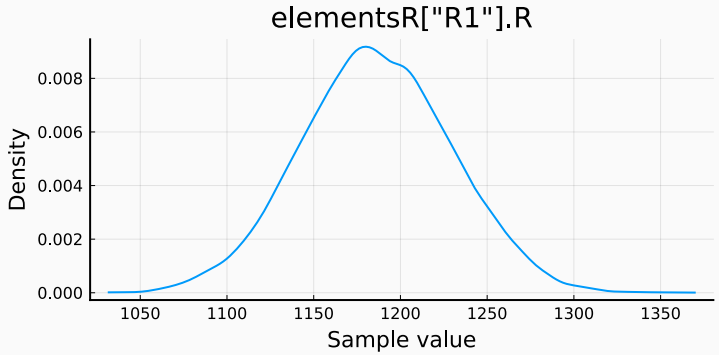


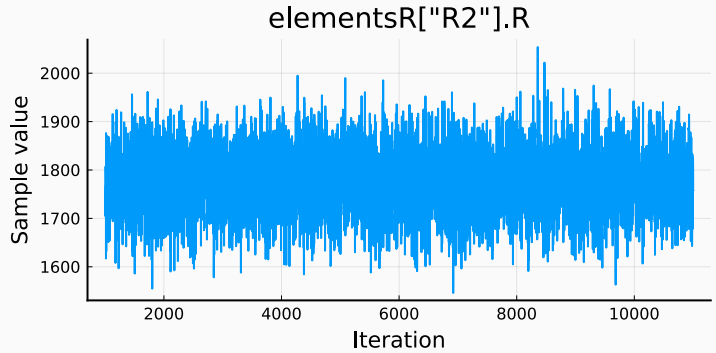
Glorified linear regression?

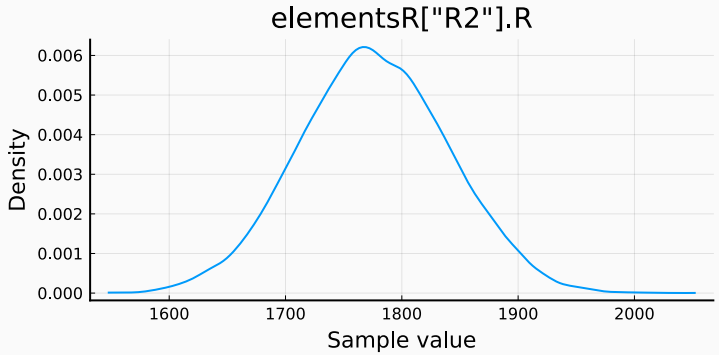




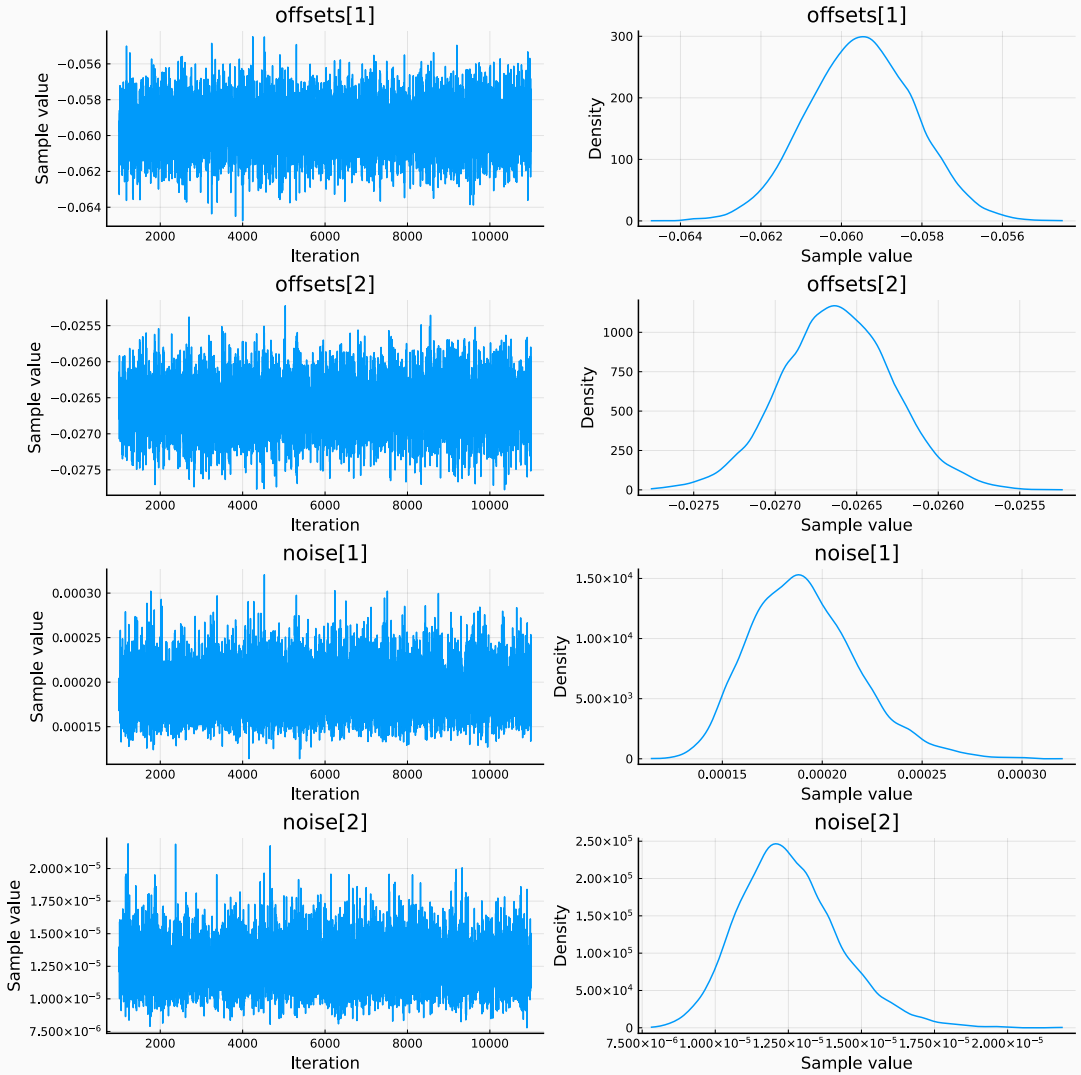




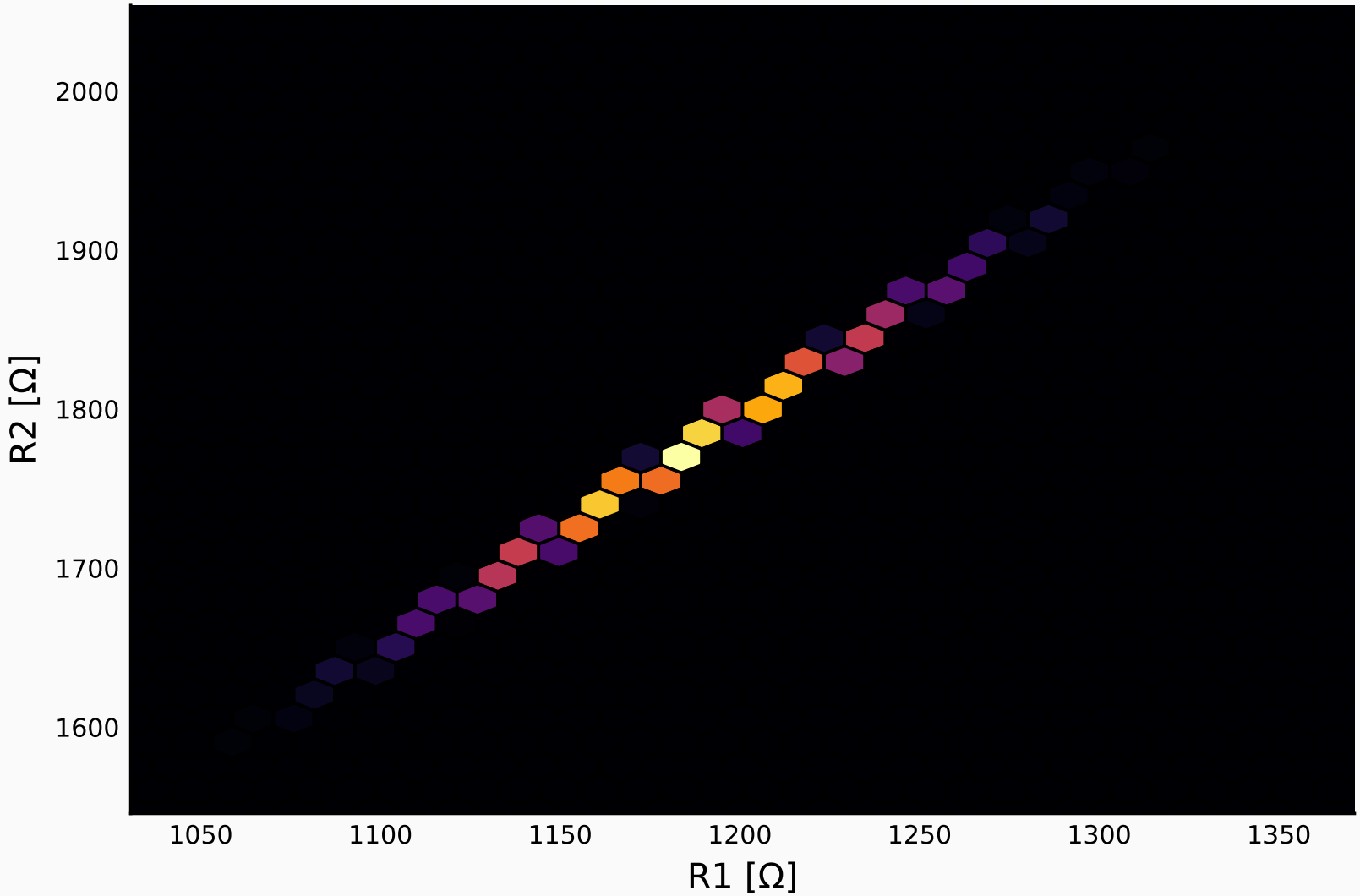




Glorified linear regression?



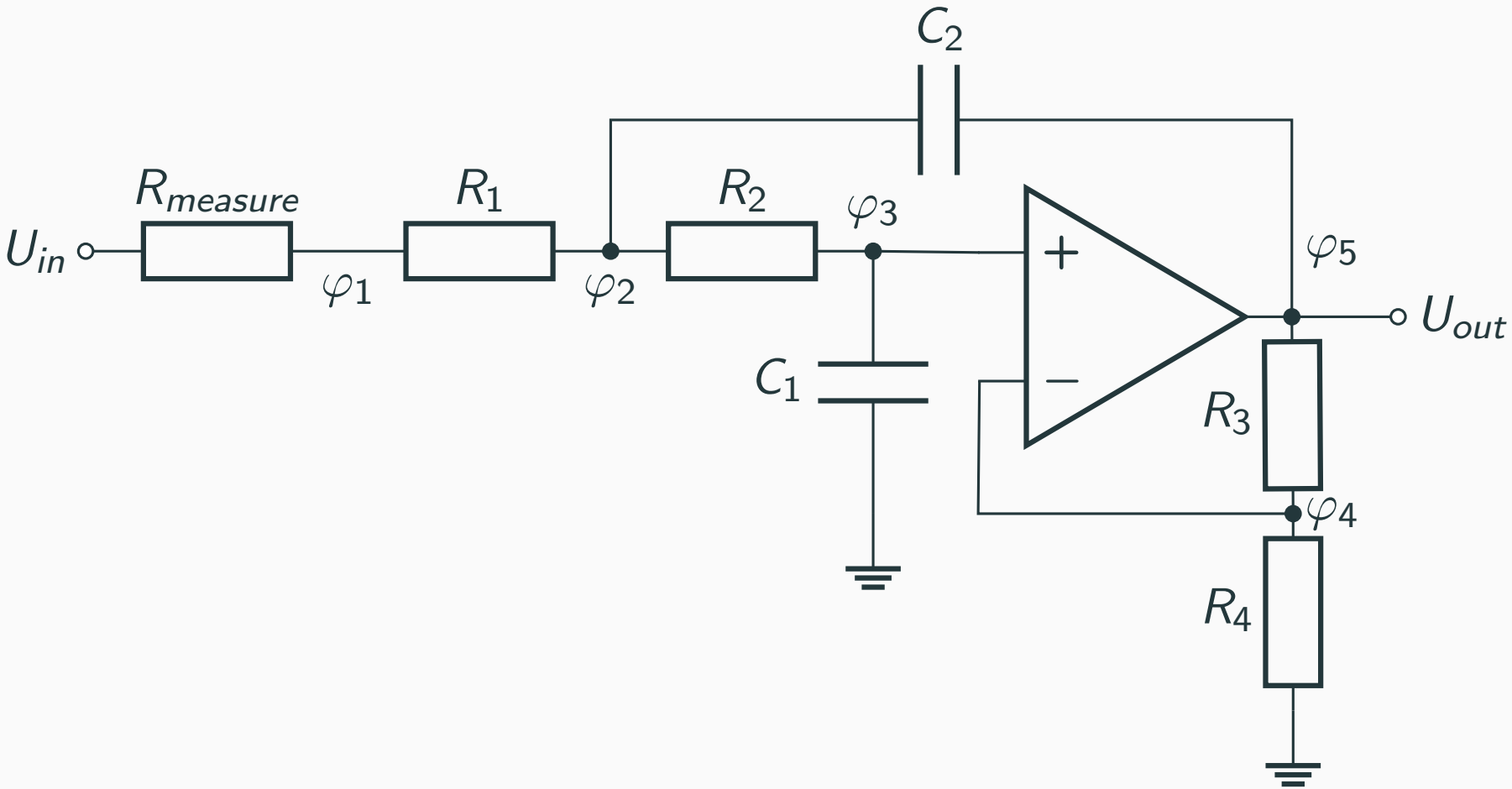
Glorified linear regression?



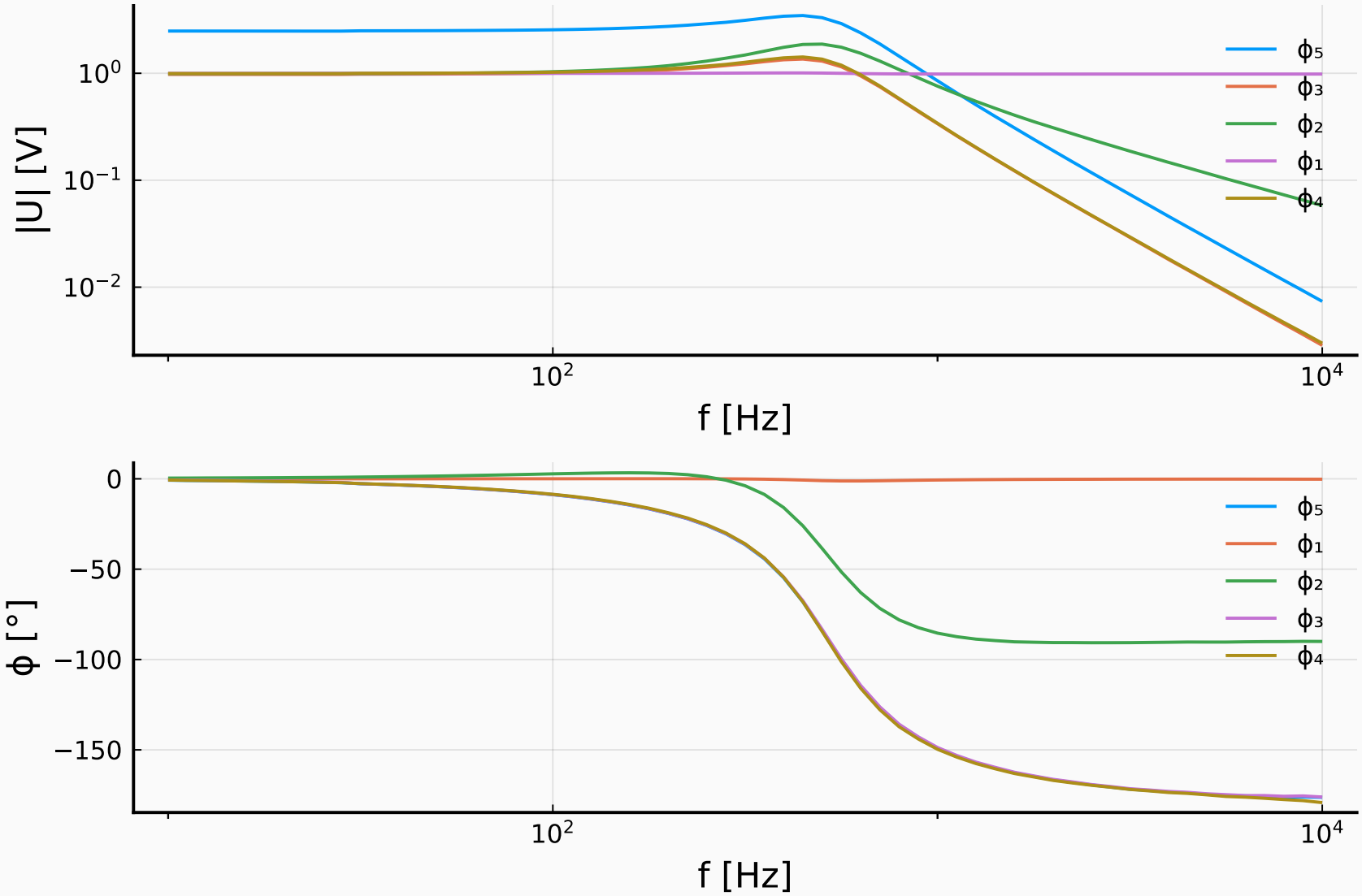
Glorified linear regression?

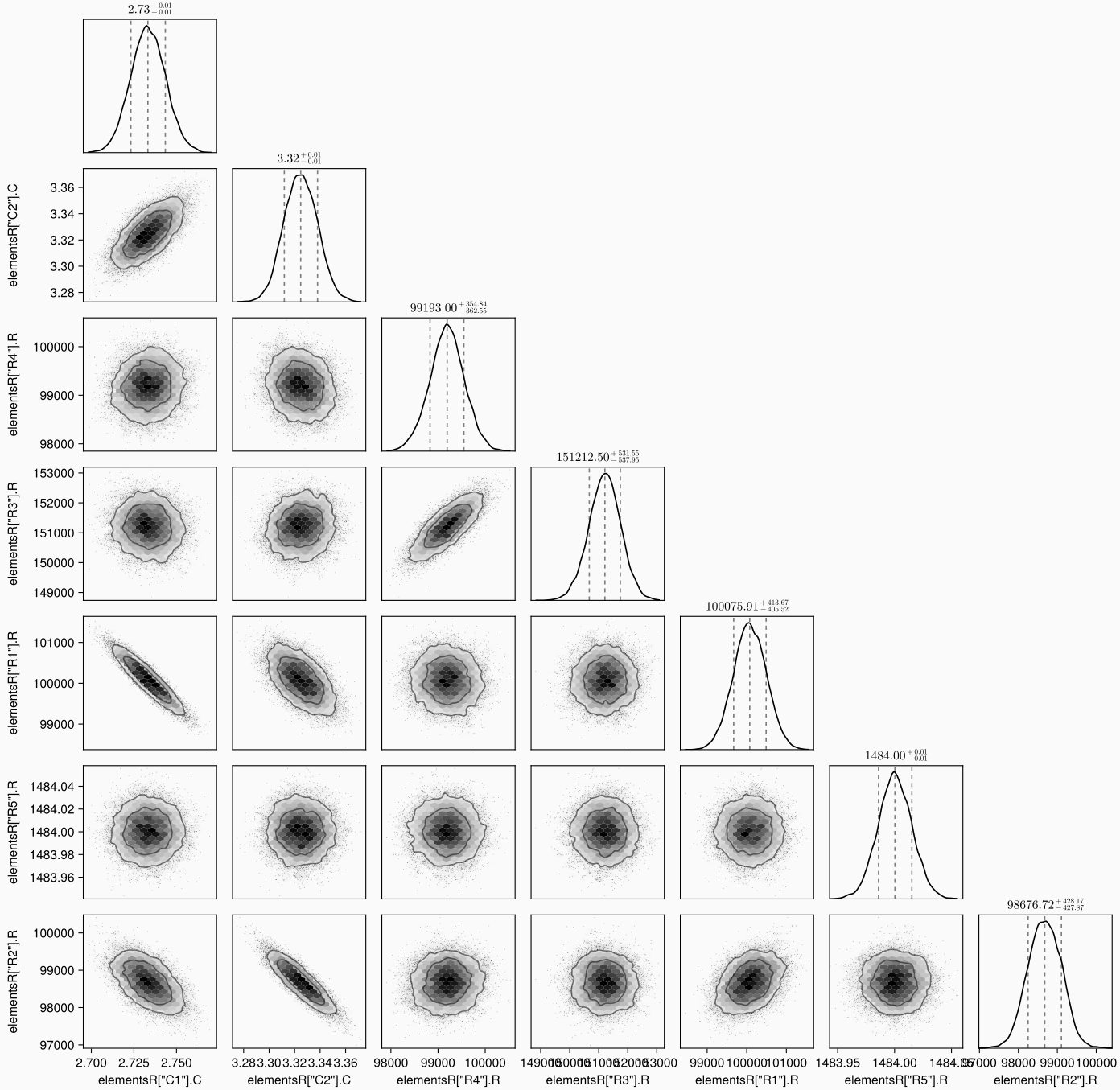
Component	Nominal Value	"True" Value	MAP-Estimate
R_1	$1.2\,\mathrm{k}\Omega$	$1.176\,\mathrm{k}\Omega$	$1.186\,\mathrm{k}\Omega$
R_2	$1.8\,\mathrm{k}\Omega$	$1.784\,\mathrm{k}\Omega$	$1.776\,\mathrm{k}\Omega$
R_3	11.72Ω	11.72Ω	11.72Ω
U_{offset_1}	?	?	$-0.059\mathrm{V}$
U_{offset_2}	?	?	$-0.026\mathrm{V}$
U_{noise_1}	?	?	$0.2m\mathrm{V}$
U_{noise_2}	?	?	$10^{-5}\mathrm{V}$

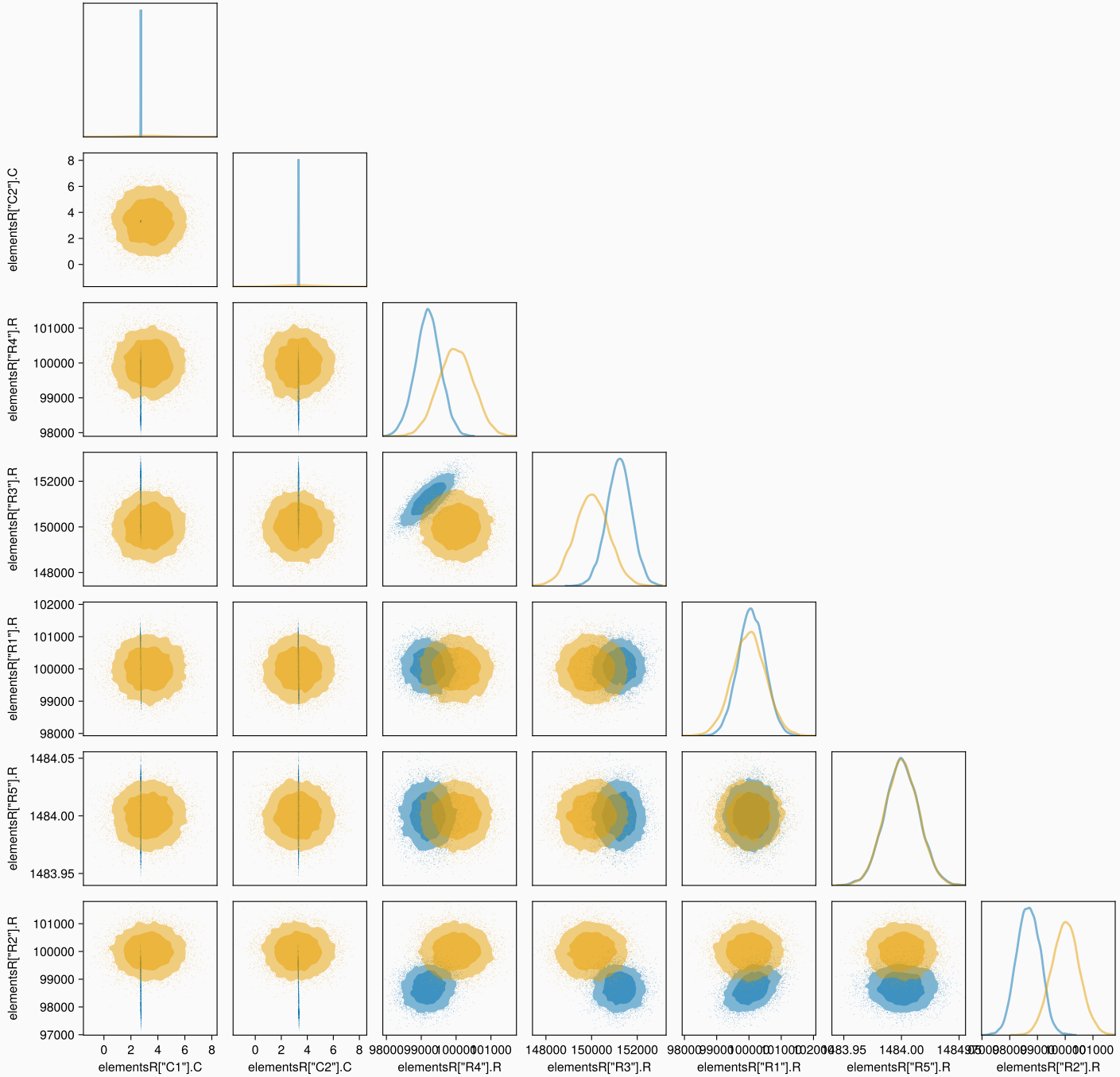
A more complex circuit



Frequency Response




$2.73^{+0.01}_{-0.01}$
$3.32^{+0.01}_{-0.01}$
$99193.00^{+354.84}_{-362.55}$
$151212.50^{+531.55}_{-537.95}$
$100075.91^{+413.67}_{-405.52}$
$1484.00^{+0.01}_{-0.01}$
$98676.72^{+428.17}_{-427.87}$
elementsR["C1"].C
elementsR["C2"].C
elementsR["R4"].R
elementsR["R3"].R
elementsR["R1"].R
elementsR["R5"].R
elementsR["R2"].R


elementsR["C2"].C
elementsR["R4"].R
elementsR["R3"].R
elementsR["R1"].R
elementsR["R5"].R
elementsR["R2"].R
elementsR["C1"].C
elementsR["C2"].C
elementsR["R4"].R
elementsR["R3"].R
elementsR["R1"].R
elementsR["R5"].R
elementsR["R2"].R

Results

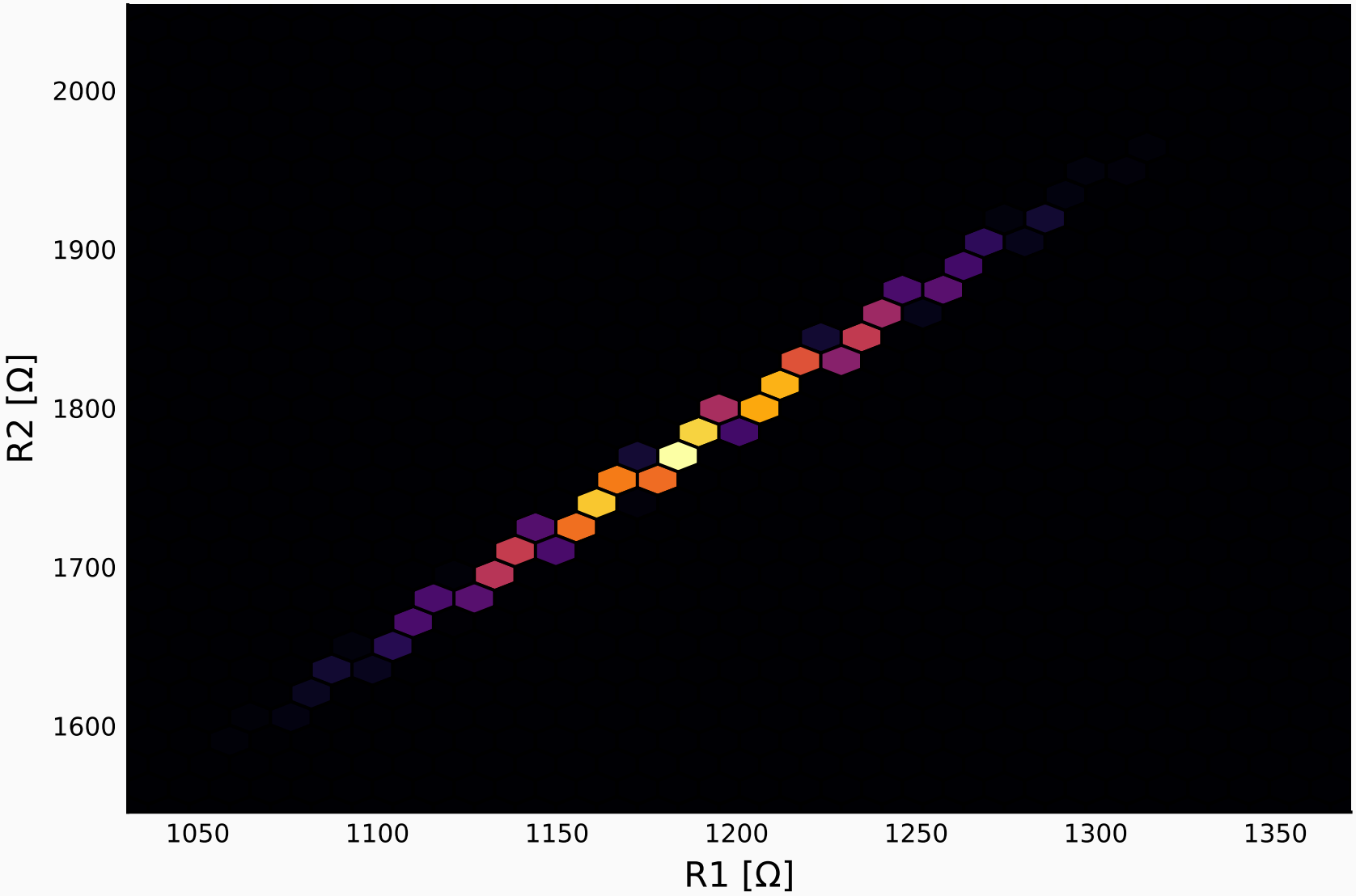
Obtained with three measured potentials: φ_1, φ_2, φ_5.

Component	Nominal Value	"True" Value	MAP-Estimate
R_1	100 kΩ	100.09 kΩ	100.08 kΩ
R_2	100 kΩ	99.97 kΩ	98.6 kΩ
R_3	150 kΩ	149.5 kΩ	151.2 kΩ
R_4	100 kΩ	100.06 kΩ	99.19Ω
C_1	3.3 nF	2.75 nF	2.73 nF
C_2	3.3 nF	3.33 nF	3.32 nF

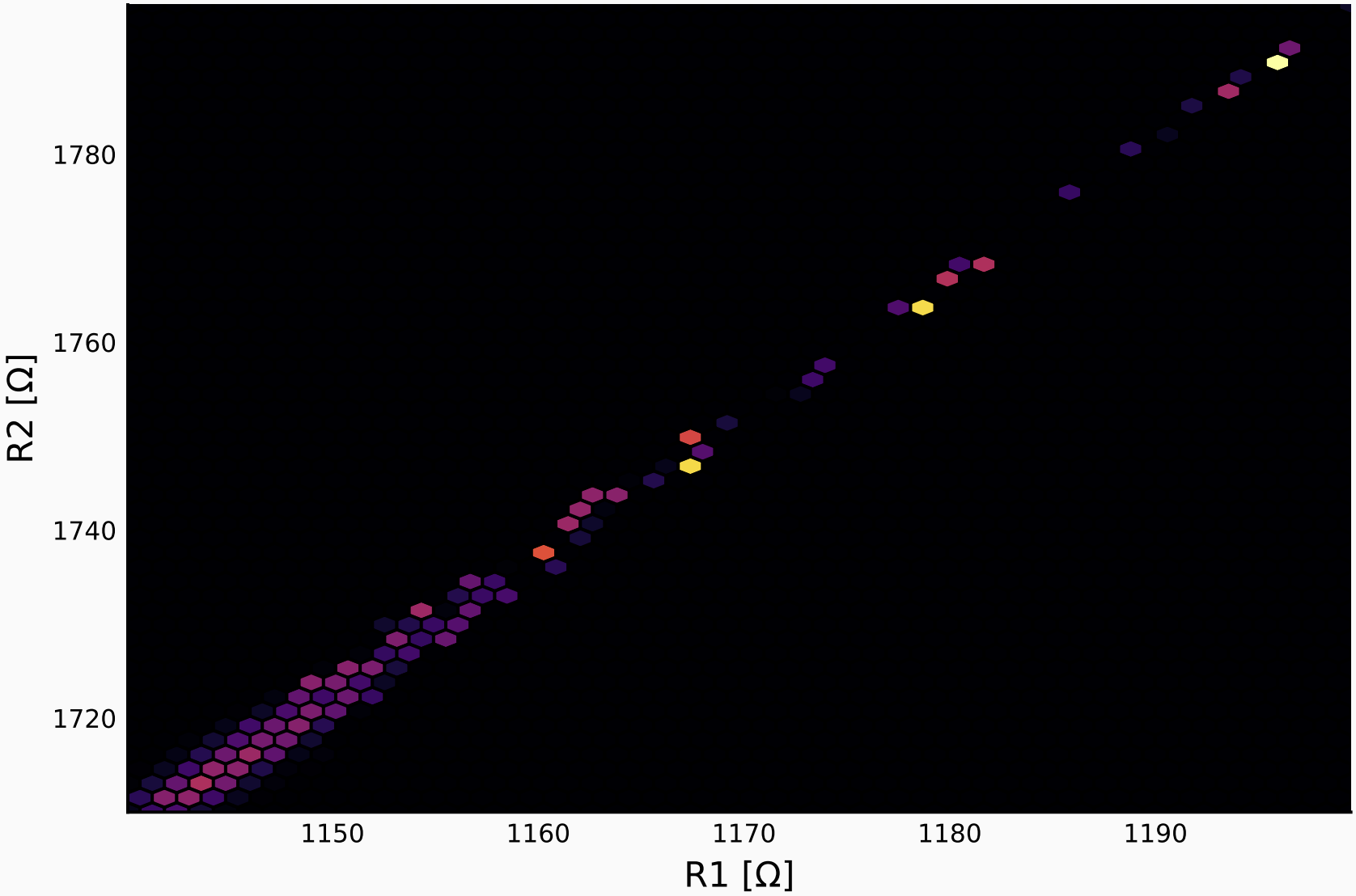
Inference

- Turing NUTS sampler performs really well
- Also tried other samplers (SMC, PG, Gibbs) but they did not work reasonably well (without further tuning)
- ADVI works wonders for this application (MAP estimate in seconds)
- Choice of priors is important: often, there output depends on a ratio of variables $\rightarrow$ scaled variables might be equally likely.

Normal distributed priors



Uniform priors



Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)

Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)

Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)
- Big range of values (e.g. $10^{-12} - 10^{6}$) $\rightarrow$ does not seem to work nicely with NUTS

Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)
- Big range of values (e.g. $10^{-12} - 10^{6}$) $\rightarrow$ does not seem to work nicely with NUTS
    - Fixed by scaling variables

Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)
- Big range of values (e.g. $10^{-12} - 10^{6}$) $\rightarrow$ does not seem to work nicely with NUTS
  - Fixed by scaling variables
  - Logarithmic frequency response (standard)

Challenges

- Getting Automatic Differentiation to work was quite a hassle (types...)
- Big range of values (e.g. $10^{-12} - 10^{6}$) $\rightarrow$ does not seem to work nicely with NUTS
  - Fixed by scaling variables
  - Logarithmic frequency response (standard)
- Hard to know if results are actually reasonable if you don't know the "true" values

Conclusion

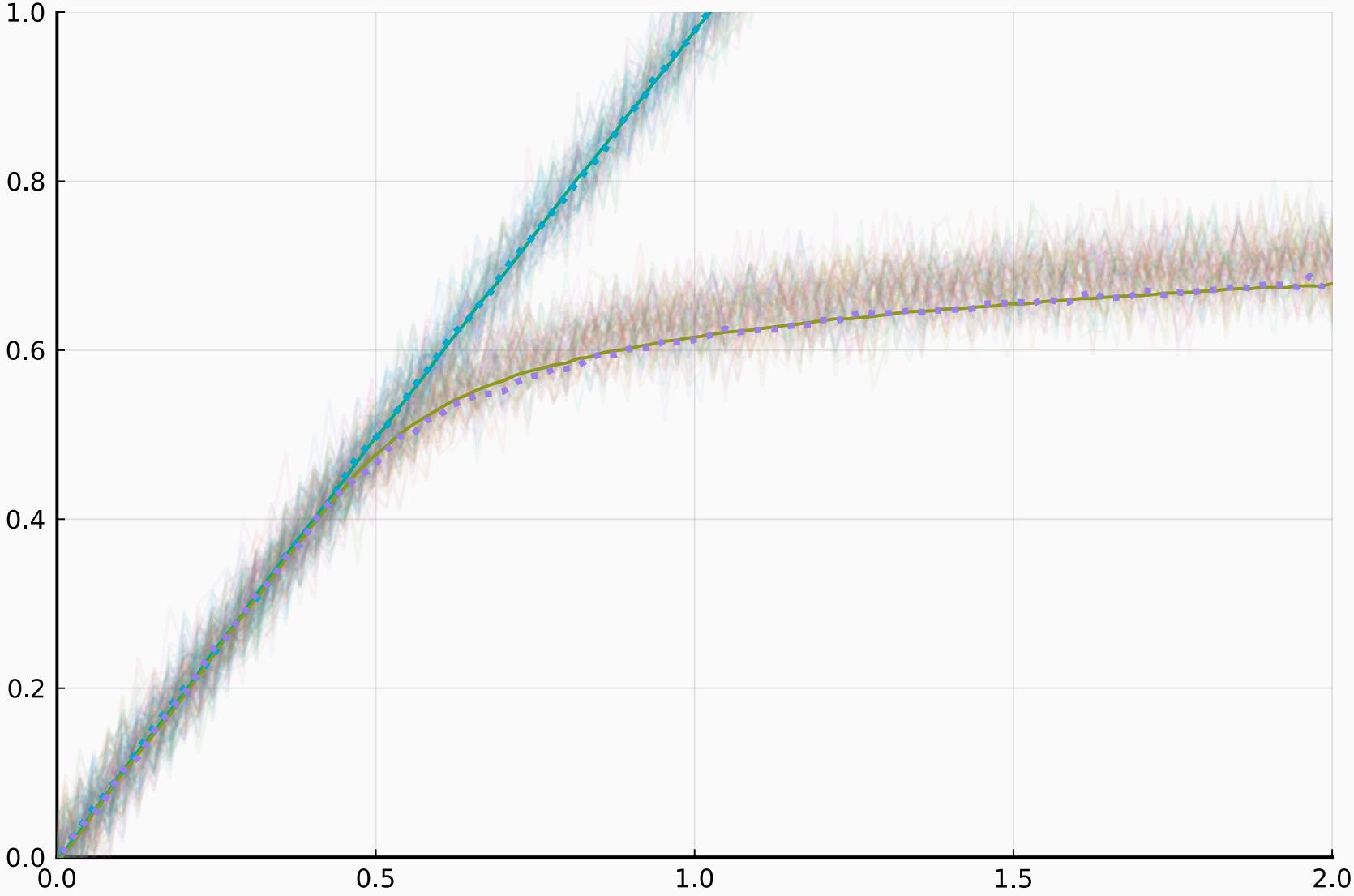
- **It works!**
- Modelling needs to account many things to be realistic (Measurement device offsets, noise, parasitics, ...)
- The nonlinear system suffers from convergence issues with some parameters explored by inference
- More components: Transistors (BJT, MOSFETs, ...)
- Transient analysis

Thank you for your attention!

Questions?

git: https://github.com/Caesiumhydroxid/SBICE

Nonlinear Diode



DC Analysis Model

```
1  @model function dcanalysis(...)
2    for e in keys(tolerances) ...
3        elementsR[e].R ~ Normal(elements[e].R,
4                    elements[e].R * tolerances[e])
5    end
6    offsets ~ MvNormal(zeros(observedNodeIndicesLength),
7                       diagm(0.01.*ones(observedNodeIndicesLength)))
8    for i in 1:observedNodeIndicesLength
9        noise[i]   ~ Uniform(0, 0.01)
10   end
11   A,RHS = assembleMatrixAndRhs(elementsR,nodes,type)
12   Ainv = inv(A)
13   for i in 1:amountOfMeasurements
14       RHS[nodes[elementsR["V1"].name]-1] = -voltages[i]
15       res = Ainv*RHS
16       results[i,:] = res[observedNodeIndices]
17   end
18   for (i,col) in enumerate(eachcol(results))
19       measurementMatrix[:,i] ~ MvNormal(col.+offsets[i],
20              diagm(noise[i]*ones(amountOfMeasurements)))
21   end
22 end
```

Sallen Key Input

```
V1 0 in AC 1
R5 in 1 1.484k
R1 1 2 100k
R2 2 3 100k
C1 2 out 3.3n
C2 3 0 3.3n
R3 out 4 150k
R4 4 0 100k
O1 3 4 out 0 1e5
```

```
R5 0.001%
R1 0.5%
R2 0.5%
R3 0.5%
R4 0.5%
C1 40%
C2 40%
```

Sallen Key Measurements

```
in Sa,Frequency in Hz,Gain in dB,Phase in °,Amplitude in Vpp
1.00E+00,1.000E+01,-2.426E-02,2.693E-03,2.500E+00
2.00E+00,1.023E+01,-2.448E-02,3.685E-03,2.500E+00
3.00E+00,1.047E+01,-2.481E-02,6.256E-04,2.500E+00
4.00E+00,1.072E+01,-2.359E-02,1.049E-02,2.500E+00
5.00E+00,1.096E+01,-2.475E-02,1.010E-02,2.500E+00
6.00E+00,1.122E+01,-2.326E-02,4.135E-03,2.500E+00
7.00E+00,1.148E+01,-2.272E-02,1.164E-02,2.500E+00
8.00E+00,1.175E+01,-2.387E-02,-7.126E-03,2.500E+00
9.00E+00,1.202E+01,-2.171E-02,1.350E-03,2.500E+00
...
```