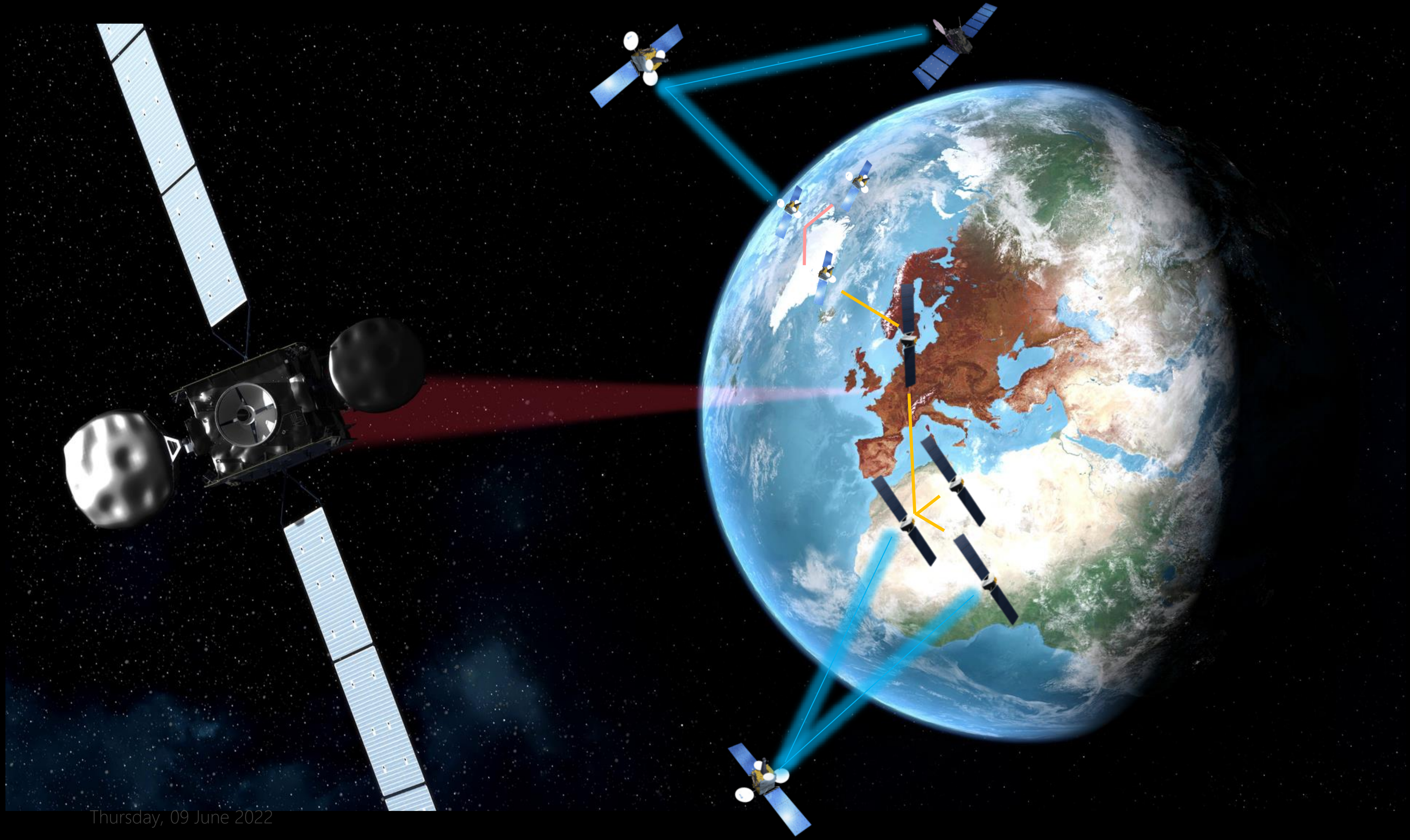
Kriging Optimisation of Antenna Elements for RF Satellite-to-Satellite Communications



Optimization methods for engineering problems (01RGBRV)

31st of May 2022

Alessio Mascolini
Andras Holczer
Gabriel Maiolini Capez

Politecnico di Torino

Who are we?

PhD students in Electrical, Electronics, and Communications Engineering and Control and Computer Engineering

Agenda

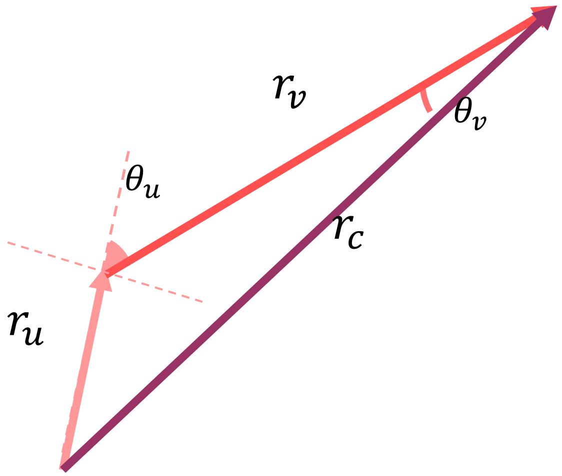
1. Problem Summary
2. Kriging
3. Kriging Surrogate Optimisation
4. Optimisation Results
5. Q&A

Problem Summary

Providing space-based broadband low-latency services to low-Earth orbit ***space*** users

- Current solutions (Ground Station Networks and Data Relay Systems) suffer from ***high data latency*** AND/OR ***limited capacity***

- Services for sea, air, land. Why not space?



Geometry of the Visibility Problem

Is the satellite visible?

Channel is usually Line-of-Sight (LOS):

- High propagation losses
- Minimum elevation angle $\theta_e \geq \theta_{min}$
- *Antenna* beamwidth, gain, steering capabilities (beamforming/steering/switching)

Moving to the Antenna…

Space Antenna Requirements in a Nutshell

1. Circular Polarisation (Axial Ratio < 3-dB)
2. 8-dBi @ 10 W in uplink (14.0-14.5 GHz)
3. 3-dBi in downlink (10.7-12.7 GHz)
4. Half-power beamwidth (HPBW) >= 100-deg
5. S11 < -10 dB
6. Competitive size, weight, and power

Single-beam *or Multi-beam*

Active, ***passive****, or hybrid*

Aperture, ***patch****, or wire*

Single element or ***array***

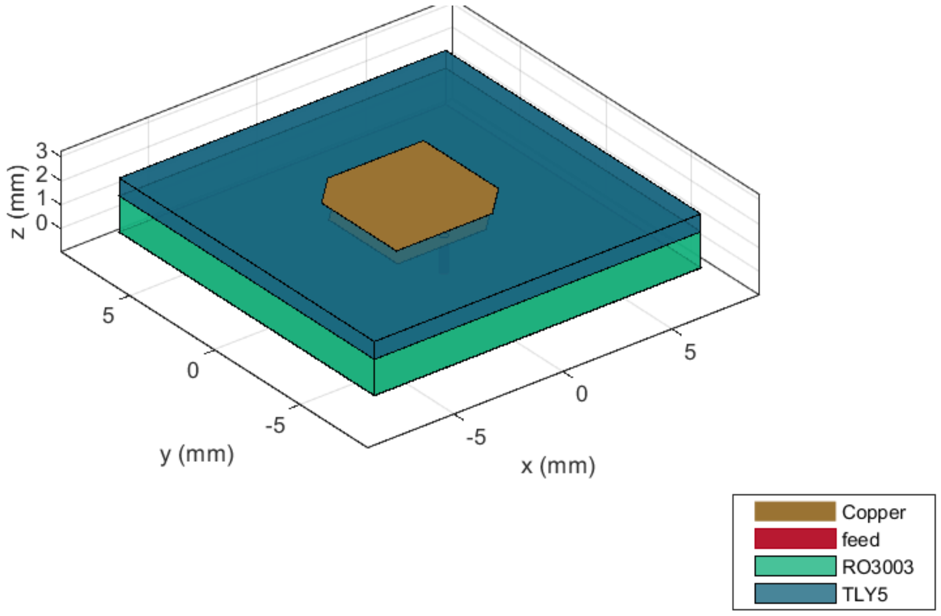
= low-cost, small footprint (~10 cm), lightweight (~100-500 g) solution

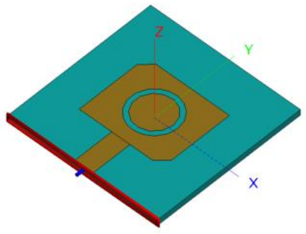
Today: Antenna Element Kriging Optimisation

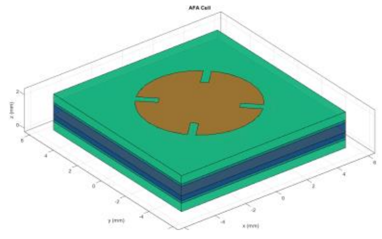
Antenna element is the *basis* of the array

- *Probe-fed Truncated Stacked Patch*
  - Driven-patch (RO3003) + Parasitic Patch (TLY5)
- Single Goal: Realised Gain >= 6-dB
  - Condenses redundant objectives (S11, AR)
- *Five* parameters to optimise:
  - *Two* Substrate Heights (coupling, inter-patch distance) ([0.127, 1.5] mm)
  - *Two* Patch Sizes ([5.0, 7.0] mm)
  - Probe Position ([-1.0, -2.0] mm)

✓ Acceptable simulation time per iteration (minutes) for a *proof-of-concept optimiser*

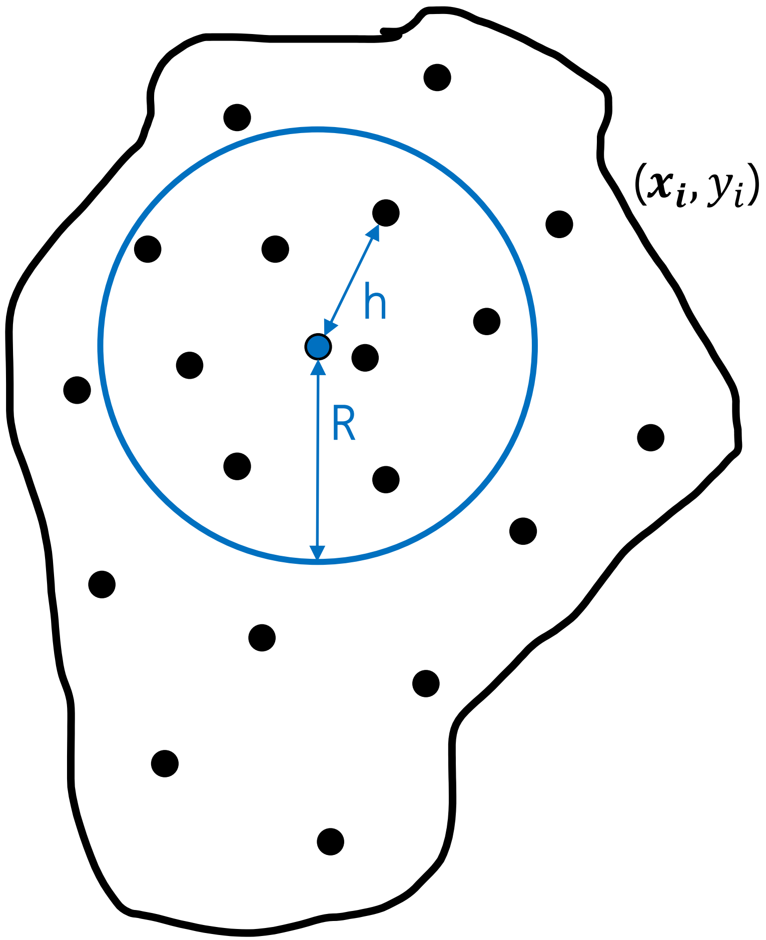






What is Kriging?

Kriging



Dataset for Kriging.

- Interpolation technique widely applied in spatial analysis.
- Starting from known points, a new point y_{new} is estimated as the linear weighed combination using surrounding values.

$$y_{new} = \sum_{i=1}^{n} \omega_i \, y_i \quad \text{, where} \quad \sum_{i=1}^{n} \omega_i = 1$$

- To determine the weighing coefficients ω_i, we can use a variogram

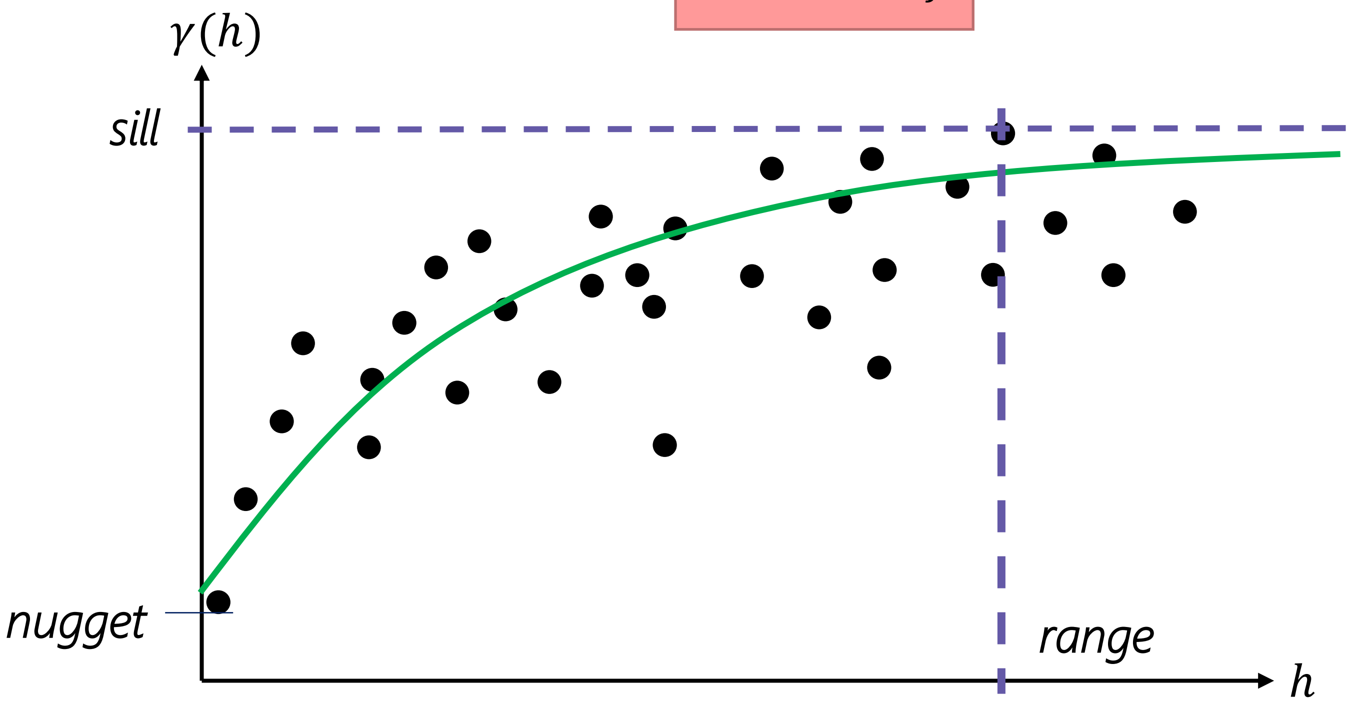
Kriging - Variogram

A variogram shows the dissimilarity of data point pairs based on their spatial distance.

- Measure of dissimilarity:

$$\gamma(\boldsymbol{x_i}, \boldsymbol{x_j}) = \frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{y_j})^2$$

$$h = \boldsymbol{x_i} - \boldsymbol{x_j}$$



Variogram and fitted parametric model

There are various parametric variogram models:

- Nugget-effect
- Bounded linear
- Spherical
- Exponential
- Gaussian
- Matérn Class

Kriging – Optimal Weight Calculation

How to compute ω_i? Many ways (simple, ordinary, universal)…

Ordinary Kriging: directly combines coefficient calculation and unbiasedness condition

$$y_{new} = \sum_{i=1}^{n} \omega_i \, y_i$$

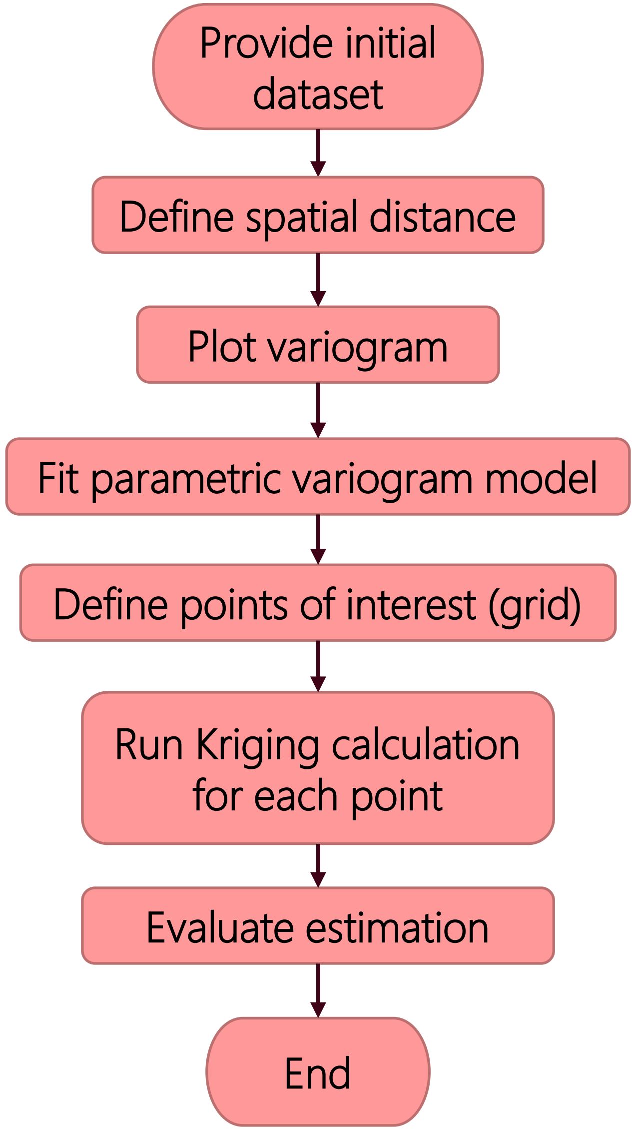
$$\sum_{i=1}^{n} \omega_i = 1$$

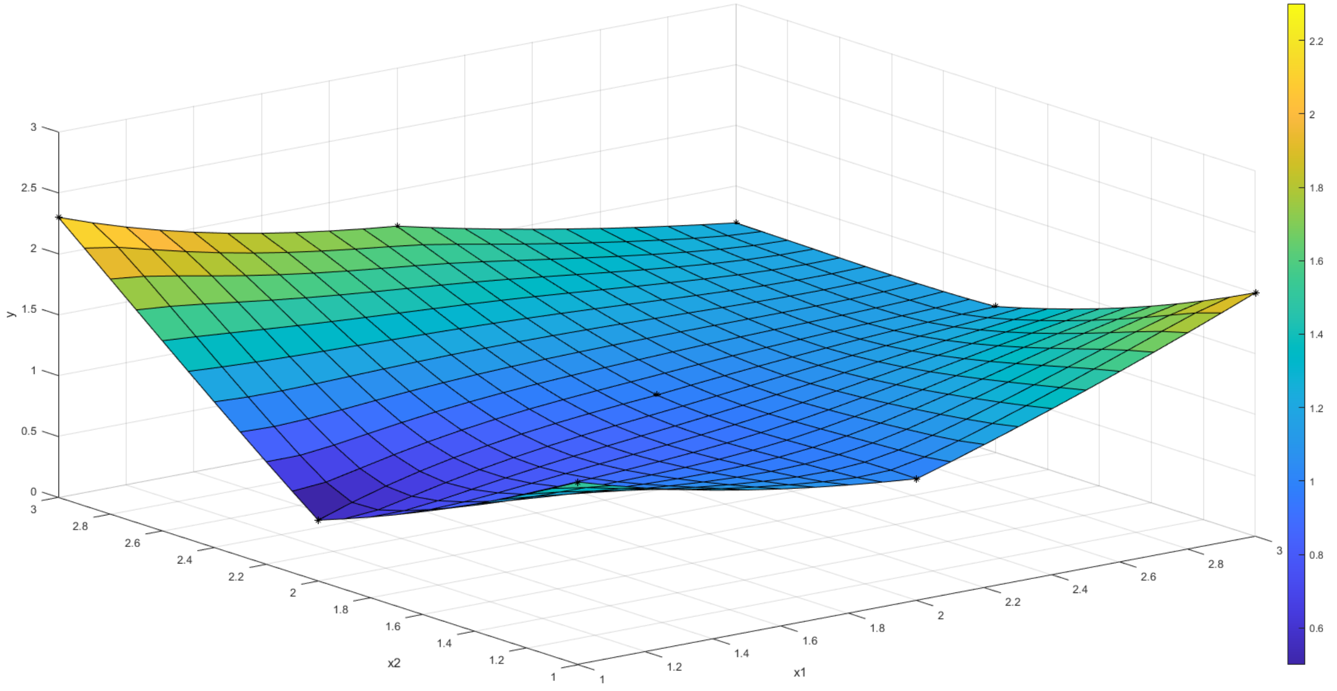
$$\begin{pmatrix} \gamma(x_1, x_1) & \gamma(x_1, x_2) & \cdots & \gamma(x_1, x_n) & 1 \\ \gamma(x_2, x_1) & \gamma(x_2, x_2) & \cdots & \gamma(x_2, x_n) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \gamma(x_n, x_1) & \gamma(x_n, x_2) & \cdots & \gamma(x_n, x_n) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \\ \lambda \end{pmatrix} = \begin{pmatrix} \gamma(x_1, x_{new}) \\ \gamma(x_2, x_{new}) \\ \vdots \\ \gamma(x_n, x_{new}) \\ 1 \end{pmatrix}$$

Also provides variance of estimation:

$$\boldsymbol{\sigma}^2 = \lambda + \sum_{i=1}^{n} \omega_i \, \gamma(x_i, x_{new})$$

Kriging Algorithm: Two Input Scalar Function Example

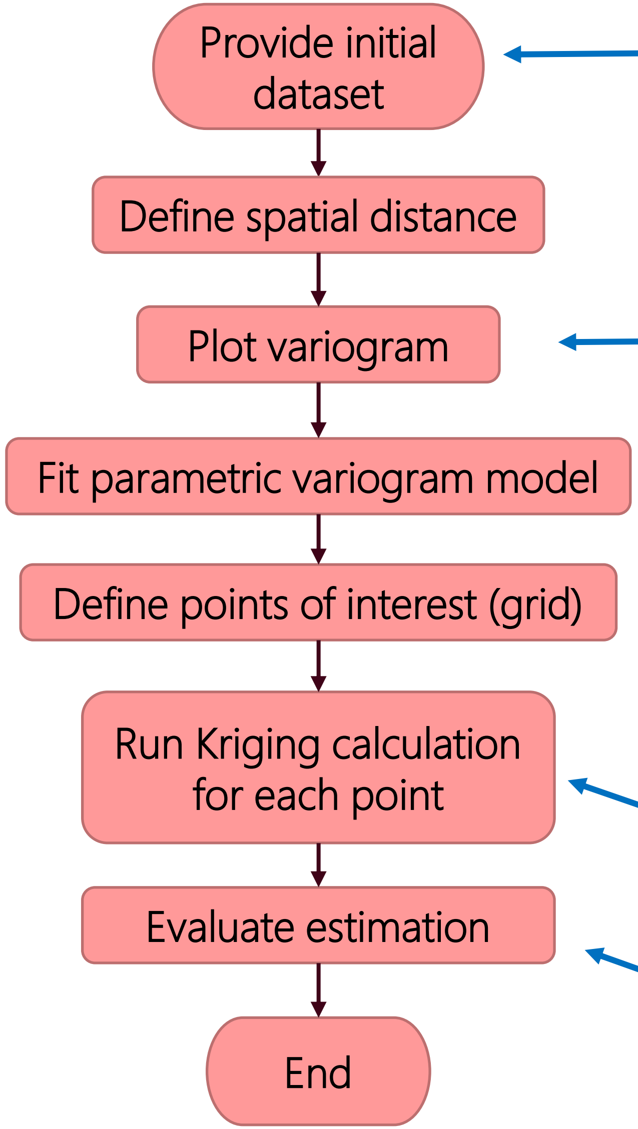




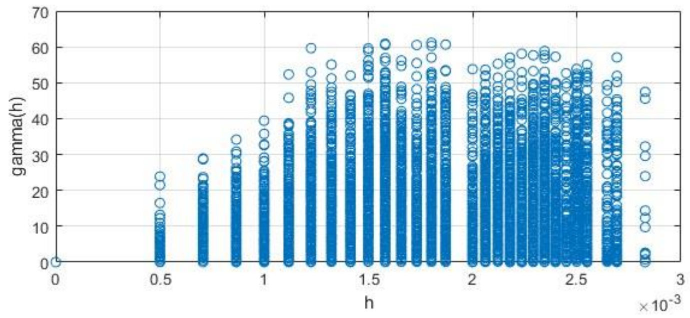
Kriging Interpolation of 2-Input Objective Function.

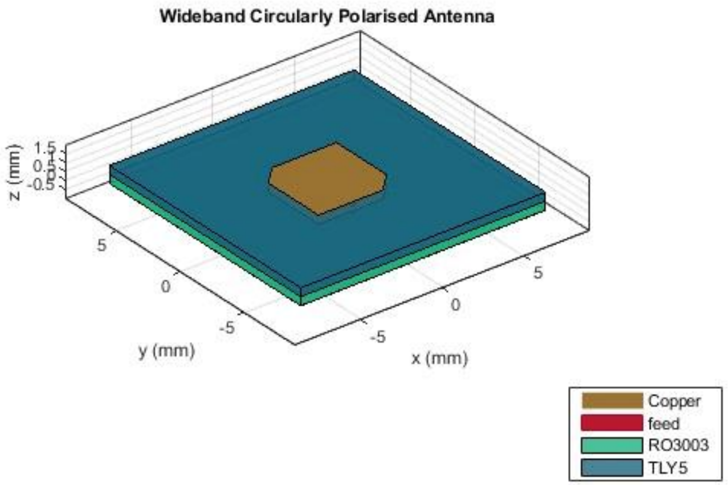
Kriging can be extended to multiple input scalar functions by properly defining the spatial distance.

Kriging Algorithm: Antenna Element with 5 Input Variables



two substrate heights (c) ([0.5, 1.5] mm)
two patch sizes
(top [5.0, 7.0] mm), bottom [6.0, 7.0] mm)
probe position ([-1.0, -2.0] mm)
Eval.: N=2 --> t=6min, N=3 --> t=1h 45min





Initial Training Dataset

P=5 --> t=4s (1s)
P=11 --> t=2min 30s (16s)
P=21 --> t=58min (380s)

Results is PxPxPxPxP array.

Finding maximum is straightforward.
Local surrogate optimization can be applied.

Kriging Surrogate Optimisation

Kriging Surrogate Optimisation

Core principle: reduce / simplify the number of ***costly*** objective function evaluations

How? By approximating it by a surrogate function, simpler and cheaper, using ***Kriging***

Trade-off: efficiency x accuracy

Let's look under the hood!

Gaussian Process Regression (Kriging Model)

Model: $y(\boldsymbol{X}) = \beta + Z(\boldsymbol{X})$, $Z(\boldsymbol{X}) \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} = \mathrm{Cov}\left(\mathrm{Z}(\boldsymbol{X}^i), \mathrm{Z}(\boldsymbol{X}^j)\right) = \boldsymbol{\sigma}^{\boldsymbol{2}} \boldsymbol{R}(\boldsymbol{X}^i, \boldsymbol{X}^j)$

Where:

- $\boldsymbol{\beta}$ is a constant term
- $\boldsymbol{y}(\boldsymbol{X})$ is the exact fitness function (EFF)
- $\mathrm{Z}(\boldsymbol{X})$ is a zero-mean Gaussian stochastic process
- $\boldsymbol{\sigma}^{\boldsymbol{2}}$ is the process variance
- $\boldsymbol{X}$ is the training data-set
- $\boldsymbol{R}(\boldsymbol{X}^i, \boldsymbol{X}^j) = \prod_{k=1}^{n} \exp(-\theta_k \left\| \boldsymbol{X}_k^i - \boldsymbol{X}_k^j \right\|^{p_k})$ is a *Gaussian Kernel,* with
    - p_k = 2 *(typ.) (Euclidean distance)*
    - θ_k *is a hyperparameter*

Global Surrogate Model (GSM)

Then, for any new point $\boldsymbol{X}^* \not\subseteq \boldsymbol{X}$:

$$\hat{y}(\boldsymbol{X}^*) = \hat{\beta} + \boldsymbol{r}(\boldsymbol{X}^*)^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{1}\hat{\beta})$$

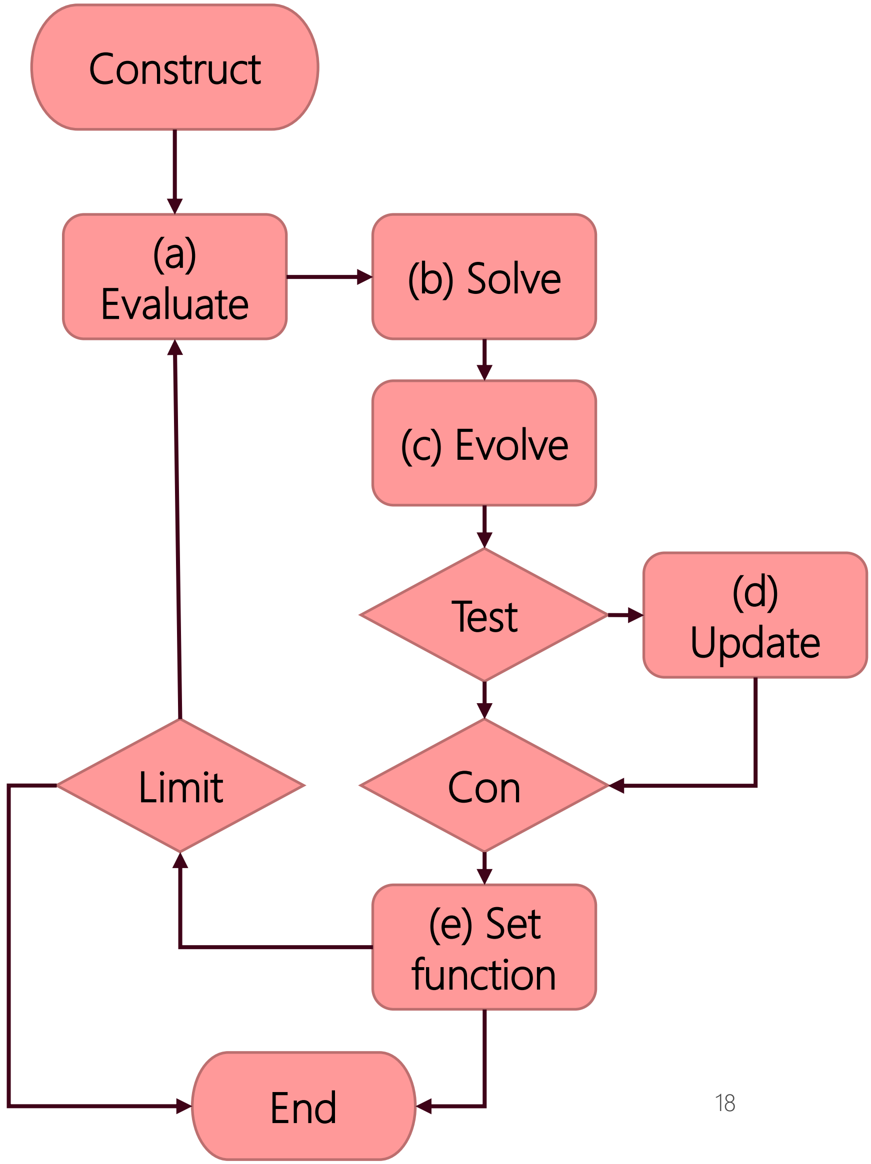
Where:

$\boldsymbol{r}(\boldsymbol{X}^*)$ is the correlation vector between the new point and those of the training data set

$\hat{y}(\boldsymbol{X}^*)$ is the global fitness function (GFF)

Hierarchical Surrogate-Assisted Evolutionary Optimization (HSAEO)

1. *Construct* a global Kriging model using available data points. *Set global* fitness function (GFF) equal to *global* surrogate model (GSM).
2. *While* iterations < limit:
   a) *Evaluate all individuals using GFF*
   b) *For each top-ranking σ-%* individuals in population (hierarchy)
      i. *Apply solver* to individual by interleaving *local* SM and *exact FF (EFF) by Kriging*
      ii. *Update database* with new points and *exact FF*
      iii. *Replace individuals* with locally improved solution (Evolutionary)
   c) *Evolve new population (Evolutionary, Genetic Algorithms)*
   d) *If GFF = EFF (Test)*
      i. *Update database* with new designs
      ii. *Update GSM* using new database
   e) If there is no improvement over Δ generations, that is, Convergence (Con), *set* GFF = EFF. *Else*, *GFF = GSM.*
   f) End.



What are Genetic Algorithms?

Genetic algorithms: starts from a set of solutions (called a population) and, by applying a set of operators (cross-over/sharing, mutation), a new population is iteratively generated and evaluated.

- Advantages:
  - Simple
  - Applicable to a wide range of optimization problems
  - Good solutions that are harder to find through other methods

- Drawback: require a *large number of function evaluations*, which is *costly* when the population size is large or the function is expensive to compute

Solution: by using a surrogate function, we can more efficiently search the solution space, improving performance (in quality and speed)

How to evolve population?

Evolutionary Strategy (deterministic) or Genetic Algorithm (probabilistic)? ***Hybrid solution***

➢ Selection-based Lamarckian-learning

Only top-ranking **σ-%** individuals evolve in intermediate generation (short-lived offspring, ***refinement***)

➢ Genetic Algorithm for New Generation

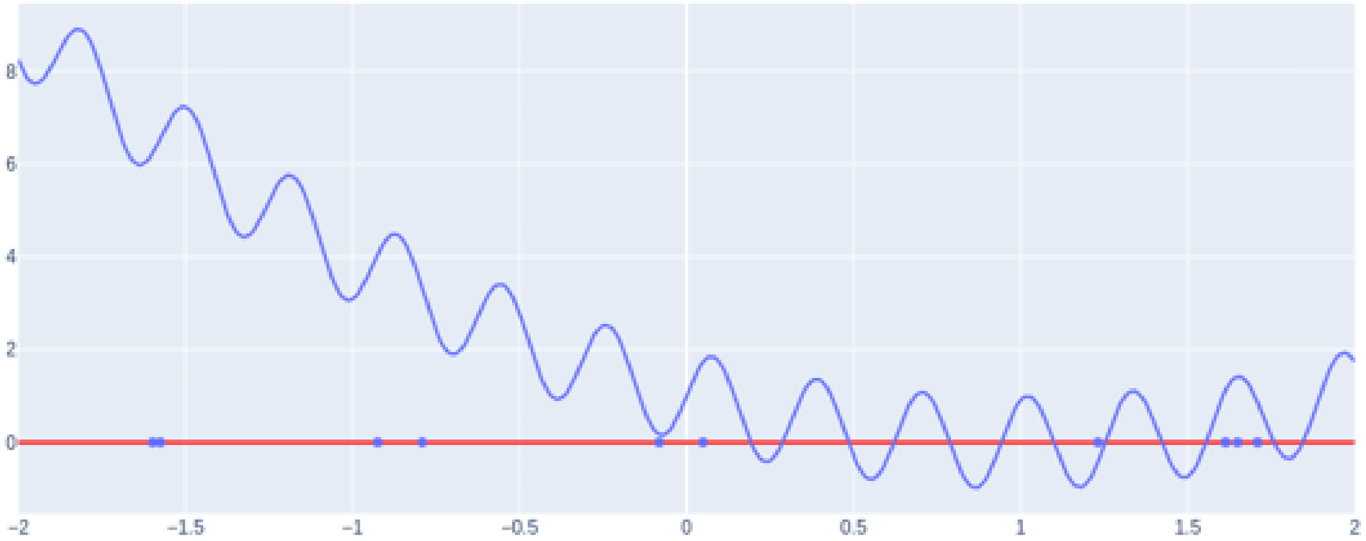
We employ a **Gaussian Regression Process** to compute the surrogate function, and **update our surrogate** as we find more suitable candidates for the **best point.**

Since this is a **stochastic model**, we can look at the standard deviation of our surrogate function. By varying its weight, we can *trade-off exploration and exploitation*

GPE Evaluation

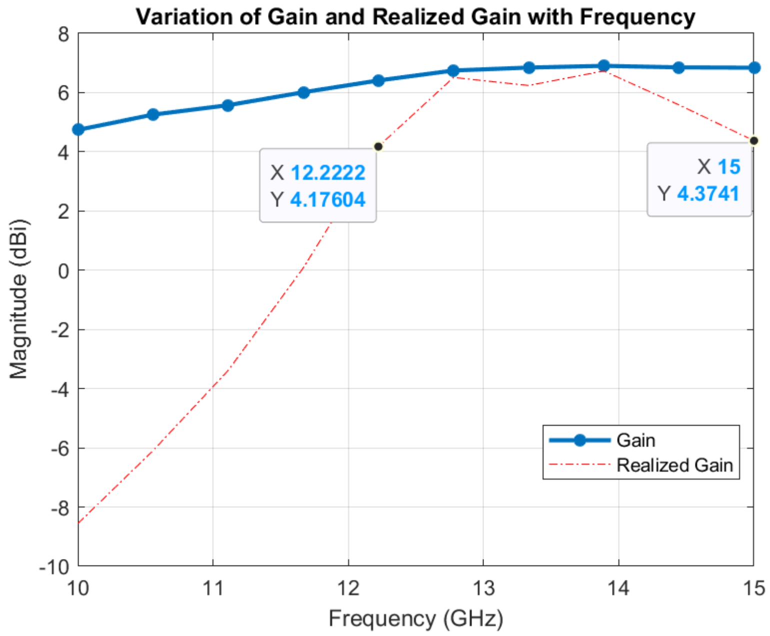
- Starting points are randomly chosen within the **bounds** and evaluated against the TFF to create the surrogate model, GP kernel: *constant + RBF*
- Top 50% of population reproduce using **crossover**
- Best points are evaluated against the **TFF** and GP is updated

Simple example: 10-to-17 function evaluations of a **1D function**

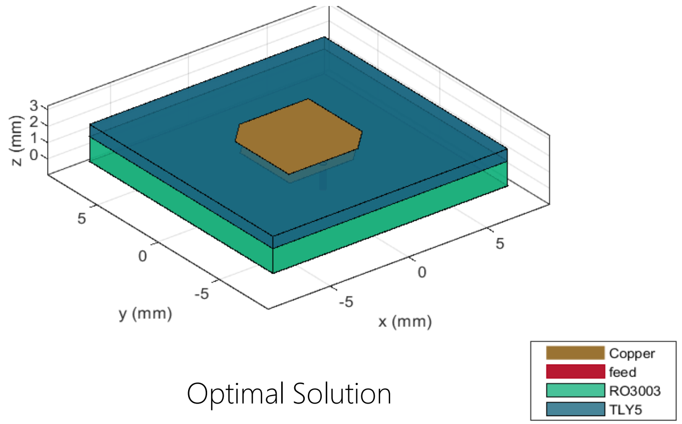


Kriging Interpolation Results

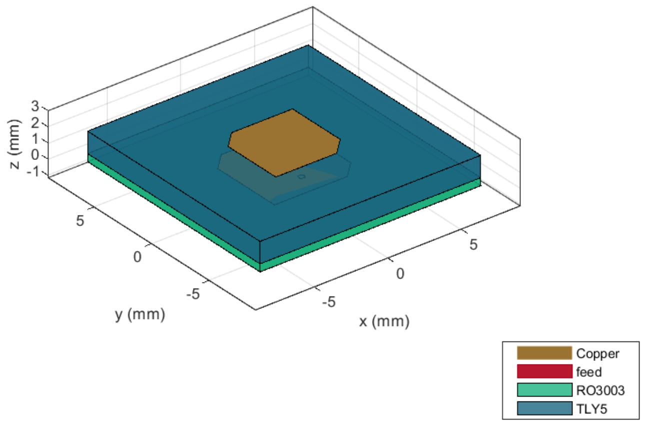
Parameter	Optimal Value
$L_{patch\ (bottom)}$	6.0 mm
$L_{patch\ (top)}$	5.2 mm
$h_{sub\ (bottom)}$	0.5 mm
$h_{sub\ (top)}$	1.4 mm
x_{feed}	1.5 mm
G_{max}	6.4 dBi



Starting Candidate

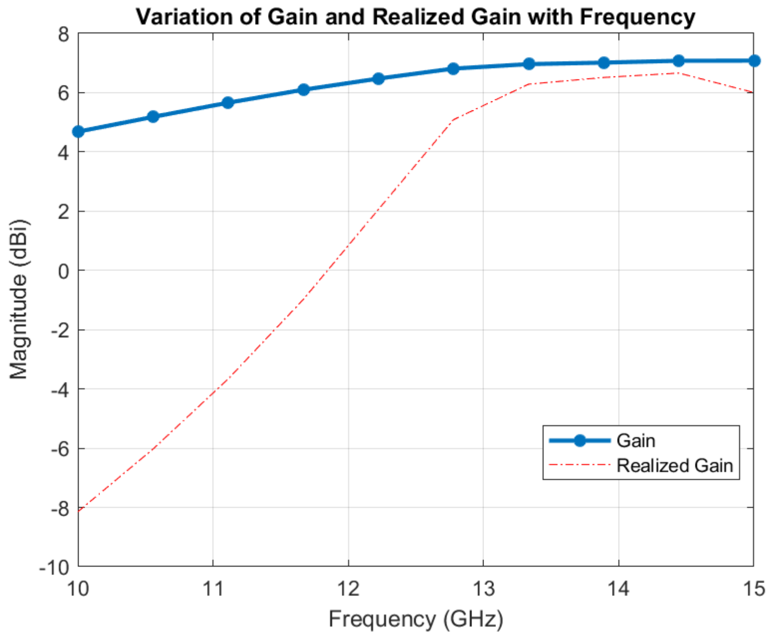


Optimal Solution

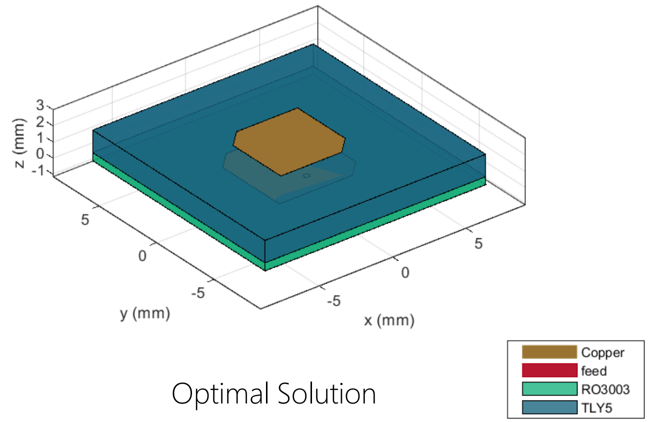


Full Optimisation Results

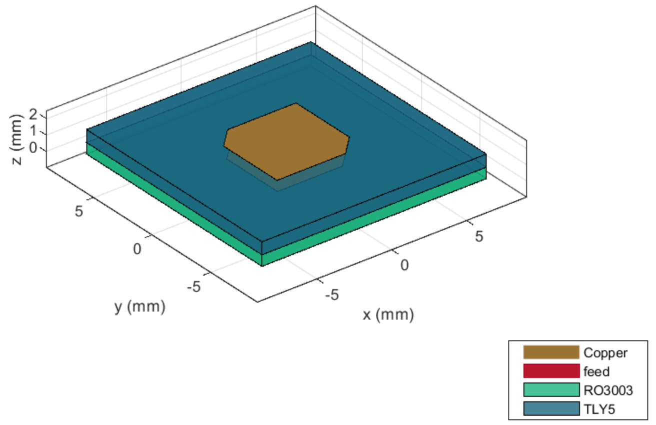
Parameter	Optimal Value
$L_{patch\ (bottom)}$	5.46 mm
$L_{patch\ (top)}$	5.81 mm
$h_{sub\ (bottom)}$	0.65 mm
$h_{sub\ (top)}$	0.80 mm
x_{feed}	1.34 mm
G_{max}	7.35 dBi



1st Kriging Candidate



Optimal Solution



Conclusion

We:

- Modelled a stacked-patch antenna element
- Implemented a proof-of-concept Kriging optimiser
- Defined a goal and set of parameters to search for a solution
- Successfully optimised the design

Questions?