initGPROptions

Initiates GPR options struct from config on GPR model and sets defaults if expected options are not available.

Syntax

```
Mdl = initGPROptions(Mdl, GPROptions)
```

Description

Mdl = initGPROptions(Mdl, GPROptions) initiates default configuration on model struct.

Input Argurments

Mdl model struct.

GPROptions options struct.

Output Argurments

Mdl model struct with attached configuration.

Requirements

- Other m-files required: None
- Subfunctions: None
- MAT-files required: None

See Also

- initGPR
- generateConfigMat


```
function Mdl = initGPROptions(Mdl, GPROptions)

    % set kernel function option
    if isfield(GPROptions, 'kernel')
        Mdl.kernel = GPROptions.kernel;
    else
        Mdl.kernel = 'QFC';
    end

    % attach hyperparameters to model and bounds for tuning and model
    % optimization
    % theta covariance function parameter theta = [s2f, sl]
    if isfield(GPROptions, 'theta')
        Mdl.theta = GPROptions.theta;
    else
        Mdl.theta = [1, 1];
    end

    % lower and upper bound for tuning theta
    if isfield(GPROptions, 's2fBounds')
        Mdl.s2fBounds = GPROptions.s2fBounds;
    else
        Mdl.s2fBounds = [1e-2, 1e2];
```

```
    end
    if isfield(GPROptions, 'slBounds')
        Mdl.slBounds = GPROptions.slBounds;
    else
        Mdl.slBounds = [1e-2, 1e2];
    end

    % noise variance s2n to predict noisy observations
    if isfield(GPROptions, 's2n')
        Mdl.s2n = GPROptions.s2n;
    else
        Mdl.s2n = 1e-5;
    end

    % lower and upper bounds for optimizing s2n
    if isfield(GPROptions, 's2nBounds')
        Mdl.s2nBounds = GPROptions.s2nBounds;
    else
        Mdl.s2nBounds = [1e-4, 10];
    end

    % enable disable mean function and correction
    if isfield(GPROptions, 'mean')
        Mdl.mean = GPROptions.mean;
    else
        Mdl.mean = 'zero';
    end

    % set polynom degree to model, default is 1 for linear correction
    if isfield(GPROptions, 'polyDegree')
        Mdl.polyDegree = GPROptions.polyDegree;

        % limit poly degree, because higher polynoms as degree 7 causes
        % an error in cholesky decomposition
        if Mdl.polyDegree > 5
            Mdl.polyDegree = 5;
        end
    else
        Mdl.polyDegree = 1;
    end

end
```

Published with MATLAB® R2020b