The Concept of the ISPoc framework

Intrusive Spectral Projection for optimal control
(preliminary version)

May 5, 2016

ISPoc is a symbolic framework for transforming uncertain optimal control problems into deterministic surrogates, which can be solved with a direct method for optimal control [1]. It is based on a symbolic implementation of the intrusive spectral projections method (Polynomial Chaos expansion) as a set of MATLAB functions. The source code for the programs can be downloaded from http://www.iwr.uni-heidelberg.de/groups/optimus/software/[2].

1 Getting started

To illustrate the functionality, we provided executable example scripts, named `run_X.m`, which show how to transform uncertain problems problems (folder `problemfiles`), how to construct different bases, moments and tensors using integration (standard) and moment-based (empirical) substitution (folder `PCE`), and how to use the postprocessing functionality (folder `postprocess`). To use the framework, change inside the main folder `ISPoc` and execute the startup script.

The source structure is as follows:

- `problemfiles`: contains model problems
  - required inputs:
    - symbolic expressions of the functions to be transformed
    - `I` - multidim. index set, e.g. `I=index_basis(length(s),K)`; for 1D polynomial order `K`, or self-selected subset
    - `s` - basis type if standard construction (per dimension/uncertain parameter) (e.g., 1 (Hermite), 2 (Legendre), 3 (Chebyshev weight)), normalized if `ext="norm"`
    - `b` - bounds per dimension, e.g. truncated normal ([] for default bounds)
  - output files are saved in subfolder `TXT`.
- `PCE`: source files implementing ISP (& empirical ISP)
- `TENSOR`: stored tensors of inner product of basis functions

2 Functionality

The core functionality of our framework consists of basis construction, tensor construction, and intrusive Galerkin projection, briefly listed in the following:

1. Basis construction:

   (i) Specification of the basis type:
   - By recursive construction using predefined weight functions: normal distribution (Hermite polynomials, Gauss weigths), uniform distribution (Legendre polynomials/weights), Chebyshev weights, other classical orthogonal polynomials can easily be added (`basis.m`)
   - Truncated normal distribution (`basis.m`)
   - By constructive generation through a symbolic Gram-Schmidt orthogonalization procedure using a given weight function or given moments (`gs_ortho.m`)

   (ii) Possibly normalize the basis polynomials (`gs_norm.m`)

[1] Tested with the software package MUSCOD-II http://www.iwr.uni-heidelberg.de/~agbock/RESEARCH/muscod.php.

[2] For remarks or questions, please write to lilli.bergner@iwr.uni-heidelberg.de

(iii) Specification of the basis order:

- Construction of a full basis up to total order p with index set

$$I_p^d = \left\{ k = (k_1, \ldots, k_d) \mid \sum_{j=1}^{d} \deg(\mathcal{P}_{k_j}) \leq p \right\}$$

generates $N_p^d = \binom{p}{p+d}$ basis functions (file `index_basis.m`)

- Construct reduced basis by selecting subset $I \subseteq I_p^d$

2. Tensor construction: (`tensor.m`)

   - Tensors are stored in a memory-saving row vector format in folder `TENSOR` for efficient reuse and evaluation.

3. Intrusive Galerkin projection of a symbolic MATLAB function:
   (file `galerkin.m`)

   - Requires symbolic functionality for simple algebraic operations (e.g., term expansion and collection) and relies on MATLAB's Symbolic Toolbox.

3 Empirical polynomial chaos **(preliminary version)**

Also termed data-driven or arbitrary polynomial chaos expansion, this variant only demands existence of a finite number of moments, which can be obtained empirically from the available data, and does not require the complete knowledge or even existence of a probability density (or weight) function. We implemented a preliminary version of the polynomial chaos transformation which replaces all integrations of the inner products of polynomials by substitution with corresponding moments.

The corresponding files are `basis_emp.m`, `gs_ortho_emp.m`, `gs_norm_emp.m`, `tensor_emp.m`. The number of required (1D) moments per uncertain parameter are $2K$ for (order K) basis generation and NK for (rank N) tensor generation.

4 Postprocessing **(preliminary version)**

ISPoc provides postprocessing functionality for the evaluation of the polynomial chaos coeffcients obtained by solving the optimal control problem. The following main function as well as an example script and help functions are located in the folder `postprocess`:

- `pce_moment.m`: calculate functional expression of moments of truncated polynomial chaos series
- `sample_state.m`: sampling of the state trajectories
- `sample_pdf.m`: sampling of the probability density function (PDF) in a histogram style and using estimators
- `plot_momentevol.m`: plot time evolution of moments of the states, additionally provides time evolution of the control policy

The input format of the obtained coefficients is adapted to the direct method MUSCOD-II but can be easily generalized, which, as well as a better documentation, is subject to a future release.