Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Nonlinear Model Order Reduction

A system-theoretic viewpoint

Maria Cruz Varona, Raphael Gebhart

Abstract

In this contribution, we consider nonlinear model order reduction from a system-theoretic viewpoint. To this end, we transfer the time domain interpretation of linear moment matching to nonlinear systems. For bilinear systems we hereby provide the time domain perception of Volterra series interpolation. For nonlinear systems we propose some simplifications to achieve a ready-to-implement, simulation-free model reduction algorithm.

Linear systems

Linear time-invariant systems

Consider a large-scale ($n \gg 10^3$) linear time-invariant (LTI) state-space model of the form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{b}\,u(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$
$$y(t) = \boldsymbol{c}^\mathsf{T}\,\boldsymbol{x}(t).$$

The input-output behavior of LTI systems is characterized in the frequency domain by the transfer function

$$G(s) = \boldsymbol{c}^\mathsf{T}(s\mathbf{I} - \boldsymbol{A})^{-1}\boldsymbol{b} = \boldsymbol{c}^\mathsf{T}\boldsymbol{\lambda}(s)\,.$$

The transfer function represents the *scaling* of a (complex) growing exponential input signal: $y(t) = G(\sigma)\,\mathrm{e}^{\sigma t}$ s.t. $u(t) = \mathrm{e}^{\sigma t}$.

Projective Model Order Reduction

The aim of model order reduction is to find a reduced order model (ROM) of much lower dimension $r \ll n$:

$$\dot{\boldsymbol{x}}_\mathrm{r}(t) = \boldsymbol{V}^\mathsf{T}\boldsymbol{A}\boldsymbol{V}\boldsymbol{x}_\mathrm{r}(t) + \boldsymbol{V}^\mathsf{T}\boldsymbol{b}\,u(t), \quad \boldsymbol{x}_\mathrm{r}(0) = \boldsymbol{V}^\mathsf{T}\boldsymbol{x}_0,$$
$$y_\mathrm{r}(t) = \boldsymbol{c}^\mathsf{T}\boldsymbol{V}\boldsymbol{x}_\mathrm{r}(t),$$

so that $y(t) \approx y_\mathrm{r}(t)$ using a *Galerkin projection* with $\boldsymbol{V}^\mathsf{T}\boldsymbol{V} = \mathbf{I}_r$.

The main task in this setting is to find a suitable (orthogonal) projection matrix $\boldsymbol{V}$. One established and numerically efficient linear reduction technique relies on the concept of implicit moment matching by *rational Krylov subspaces*.

Concept of Moment Matching

Choosing $\boldsymbol{V}$ as a basis of an input rational Krylov subspace

$$\mathrm{span}\left\{(\sigma_1\mathbf{I} - \boldsymbol{A})^{-1}\boldsymbol{b}, \ldots, (\sigma_r\mathbf{I} - \boldsymbol{A})^{-1}\boldsymbol{b}\right\} \subseteq \mathrm{Ran}(\boldsymbol{V})$$

guarantees that the ROM matches r moments of the transfer function $G(s)$ around the selected shifts $\sigma_1, \ldots, \sigma_r$.

Linear moment matching can be also interpreted in time domain as the interpolation of the output of the FOM, when this is excited by growing exponentials $u(t) = \boldsymbol{e}^\mathsf{T}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}$ with frequencies $\boldsymbol{S}_v = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ and $\boldsymbol{e}^\mathsf{T} = [1, \ldots, 1]$:

$$\left(\boldsymbol{V}\,\boldsymbol{S}_v - \boldsymbol{A}\,\boldsymbol{V} - \boldsymbol{b}\,\boldsymbol{e}^\mathsf{T}\right)\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} = \boldsymbol{0} \iff \boldsymbol{V}\,\boldsymbol{S}_v = \boldsymbol{A}\,\boldsymbol{V} + \boldsymbol{b}\,\boldsymbol{e}^\mathsf{T}.$$

Polynomial nonlinear systems

Bilinear dynamical systems

The *Carleman linearization method* allows to represent nonlinear dynamical systems by polynomial nonlinear systems through a power series expansion of the nonlinearities:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}^{(1)}\,\boldsymbol{x}(t) + \boldsymbol{A}^{(2)}\left(\boldsymbol{x}(t) \otimes \boldsymbol{x}(t)\right) + \cdots + \boldsymbol{N}^{(1)}\,\boldsymbol{x}(t)\,u(t) + \cdots + \boldsymbol{b}\,u(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$
$$y(t) = \boldsymbol{c}^\mathsf{T}\,\boldsymbol{x}(t).$$

One special class for polynomial nonlinear systems are bilinear dynamical systems:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{N}\,\boldsymbol{x}(t)\,u(t) + \boldsymbol{b}\,u(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$
$$y(t) = \boldsymbol{c}^\mathsf{T}\,\boldsymbol{x}(t).$$

Using the *Volterra series representation* for such polynomial systems enables to break them down into an infinite series of homogenous, interconnected subsystems.

Variational analysis

The *variational equation approach* allows us to obtain a state-equation for each degree-k subsystem. To this end, it is assumed that the system's response to an input of the form $\alpha u(t)$ can be written as a sum of sub-responses:

$$\boldsymbol{x}(t) = \alpha\boldsymbol{x}_1(t) + \alpha^2\boldsymbol{x}_2(t) + \alpha^3\boldsymbol{x}_3(t) \ldots$$

Inserting the assumed input and the assumed response into e.g. the bilinear system yields:

$$\alpha\dot{\boldsymbol{x}}_1(t) + \alpha^2\dot{\boldsymbol{x}}_2(t) + \ldots = \boldsymbol{A}\left(\alpha\boldsymbol{x}_1(t) + \alpha^2\boldsymbol{x}_2(t) + \ldots\right)$$
$$+ \boldsymbol{N}\left(\alpha\boldsymbol{x}_1(t) + \alpha^2\boldsymbol{x}_2(t) + \ldots\right)\alpha u(t) + \boldsymbol{b}\,\alpha u(t)\,.$$

Since this differential equation must hold for all α, terms of like powers of α can be equated, yielding a state-equation for each subsystem:

$$\alpha: \quad \dot{\boldsymbol{x}}_1(t) = \boldsymbol{A}\,\boldsymbol{x}_1(t) + \boldsymbol{b}\,u(t), \qquad \boldsymbol{x}_1(0) = \boldsymbol{x}_0,$$
$$\alpha^k: \quad \dot{\boldsymbol{x}}_k(t) = \boldsymbol{A}\,\boldsymbol{x}_k(t) + \boldsymbol{N}\,\boldsymbol{x}_{k-1}(t)\,u(t), \quad \boldsymbol{x}_k(0) = \boldsymbol{0}, \quad k \geq 2.$$

Growing Exponential Approach

This approach is used to obtain an input-output characterization for a polynomial system in terms of the scaling of a *sum of growing exponentials as input signal*: $u(t) = \sum_{l_1=1}^{r} u_{l_1}\,\mathrm{e}^{\sigma_{l_1} t}$.

For each subsystem, the assumed solution takes the form

$$\boldsymbol{x}_k(t) = \sum_{l_1=1}^{r} \cdots \sum_{l_k=1}^{r} \boldsymbol{\lambda}_k^{\triangle}(\sigma_{l_1}, \ldots, \sigma_{l_k})\,u_{l_1} \cdots u_{l_k}\,\mathrm{e}^{\sigma_{l_1} t} \cdots \mathrm{e}^{\sigma_{l_k} t}.$$

Inserting the assumed solution together with the applied input signal in each subsystem state-equation yields the respective scaling factors:

$$\boldsymbol{\lambda}_1(\sigma_{l_1}) = (\sigma_{l_1}\mathbf{I} - \boldsymbol{A})^{-1}\boldsymbol{b},$$
$$\boldsymbol{\lambda}_k^{\triangle}(\sigma_{l_1}, \ldots, \sigma_{l_k}) = \left((\sigma_{l_1} + \cdots + \sigma_{l_k})\mathbf{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{N}\boldsymbol{\lambda}_{k-1}^{\triangle}(\sigma_{l_1}, \ldots, \sigma_{l_{k-1}}), \quad k \geq 2.$$

Moment Matching by Volterra series interpolation

Choosing $\boldsymbol{V} = \sum_{k=1}^{\infty}\boldsymbol{V}^{(k)}$ as the solution of the (truncated) bilinear Sylvester equation:

$$\boldsymbol{V}^{(1)}\,\boldsymbol{S}_v = \boldsymbol{A}\,\boldsymbol{V}^{(1)} + \boldsymbol{b}\,\boldsymbol{e}^\mathsf{T},$$
$$\boldsymbol{V}^{(k)}\,\boldsymbol{S}_v = \boldsymbol{A}\,\boldsymbol{V}^{(k)} + \boldsymbol{N}\,\boldsymbol{V}^{(k-1)}\,\boldsymbol{U}_v^\mathsf{T}, \quad k \geq 2,$$

with shifts $\boldsymbol{S}_v = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ and weights $\boldsymbol{U}_v$, guarantees Volterra series interpolation.

From a time domain perspective, the question raises how the input is being implicitly chosen to achieve Volterra series interpolation. In fact, inserting the ansatz $\boldsymbol{x}_k(t) = \boldsymbol{V}^{(k)}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}$ in each subsystem state-equation yields:

$$\boldsymbol{V}^{(1)}\,\boldsymbol{S}_v\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} = \boldsymbol{A}\,\boldsymbol{V}^{(1)}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} + \boldsymbol{b}\,u(t),$$
$$\boldsymbol{V}^{(k)}\,\boldsymbol{S}_v\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} = \boldsymbol{A}\,\boldsymbol{V}^{(k)}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} + \boldsymbol{N}\,\boldsymbol{V}^{(k-1)}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}\,u(t), \quad k \geq 2.$$

To enforce Volterra series interpolation, the input has to fulfill the following conditions:

$$u(t) \stackrel{!}{=} \boldsymbol{e}^\mathsf{T}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0} \quad \text{and} \quad \boldsymbol{V}^{(k-1)}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}\,u(t) \stackrel{!}{=} \boldsymbol{V}^{(k-1)}\,\boldsymbol{U}_v^\mathsf{T}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}.$$

Nonlinear systems

Nonlinear dynamical systems

Consider now a large-scale, nonlinear time-invariant, exponentially stable, state-space model of the form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t), u(t)\big), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$
$$y(t) = h\big(\boldsymbol{x}(t)\big).$$

In order to find a nonlinear ROM, one could apply a *nonlinear Petrov-Galerkin projection* with nonlinear mappings $\boldsymbol{x}(t) = \boldsymbol{\nu}(\boldsymbol{x}_\mathrm{r}(t))$ and $\boldsymbol{x}_\mathrm{r}(t) = \boldsymbol{\omega}(\boldsymbol{x}(t))$ such that $\boldsymbol{\omega}(\boldsymbol{\nu}(\boldsymbol{x}_\mathrm{r})) = \boldsymbol{x}_\mathrm{r}$. Another easier and established way consists in applying the classical linear projection ansatz $\boldsymbol{x}(t) = \boldsymbol{V}\,\boldsymbol{x}_\mathrm{r}(t)$ and $\boldsymbol{x}_\mathrm{r}(t) = \boldsymbol{W}^\mathsf{T}\,\boldsymbol{x}(t)$, where $\boldsymbol{W} = \boldsymbol{V}$ is commonly used, yielding $\boldsymbol{V}^\mathsf{T}\boldsymbol{V} = \mathbf{I}_r$.

Notion of signal generators

The concept of moment matching can be transferred to nonlinear systems using *signal generators*. In fact, interconnecting a system with the following *linear* signal generator:

$$\left.\begin{aligned} \dot{\boldsymbol{x}}_\mathrm{r}(t) &= \boldsymbol{S}_v\,\boldsymbol{x}_\mathrm{r}(t), \quad \boldsymbol{x}_\mathrm{r}(0) = \boldsymbol{x}_{\mathrm{r},0}, \\ u(t) &= \boldsymbol{e}^\mathsf{T}\,\boldsymbol{x}_\mathrm{r}(t), \end{aligned}\right\} \implies \left\{\begin{aligned} \boldsymbol{x}_\mathrm{r}(t) &= \mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}, \\ u(t) &= \boldsymbol{e}^\mathsf{T}\,\mathrm{e}^{\boldsymbol{S}_v t}\,\boldsymbol{x}_{\mathrm{r},0}, \end{aligned}\right.$$

corresponds to exciting the system with a sum of growing exponentials as input signal.

Practicable simulation-free nonlinear moment matching

A linear projection ansatz $\boldsymbol{x}(t) = \boldsymbol{V}\,\boldsymbol{x}_\mathrm{r}(t)$ together with a linear signal generator for the nonlinear system yields the following state-dependent nonlinear system of equations (NLSE):

$$\boldsymbol{0} = \boldsymbol{f}\big(\boldsymbol{V}\boldsymbol{x}_\mathrm{r}(t),\, \boldsymbol{e}^\mathsf{T}\,\boldsymbol{x}_\mathrm{r}(t)\big) - \boldsymbol{V}\,\boldsymbol{S}_v\,\boldsymbol{x}_\mathrm{r}(t), \ \forall\,\boldsymbol{x}_\mathrm{r}(t),$$

with user-defined $(\boldsymbol{S}_v, \boldsymbol{e}^\mathsf{T}, \boldsymbol{x}_{\mathrm{r},0})$. This underdetermined NLSE can be considered column-wise. Since the state vector $\boldsymbol{x}_\mathrm{r}(t)$ could not be factored out, we propose to discretize the state-dependent NLSE with time-snapshots $\{t_k^*\}$, $k = 1, \ldots, K$ similar as POD:

$$\boldsymbol{0} = \boldsymbol{f}\big(\boldsymbol{v}_{ik}\,x_{\mathrm{r},i}(t_k^*),\, x_{\mathrm{r},i}(t_k^*)\big) - \sigma_i\,\boldsymbol{v}_{ik}\,x_{\mathrm{r},i}(t_k^*),$$

with $x_{\mathrm{r},i}(t_k^*) = \mathrm{e}^{\sigma_i t_k^*}\,x_{\mathrm{r},0,i}$ for $i = 1, \ldots, r$.

Alternatively, one could also apply a *nonlinear* signal generator that is chosen accordingly:

$$\dot{\boldsymbol{x}}_\mathrm{r}(t) = \boldsymbol{s}_v\big(\boldsymbol{x}_\mathrm{r}(t)\big), \quad \boldsymbol{x}_\mathrm{r}(0) = \boldsymbol{x}_{\mathrm{r},0},$$
$$u(t) = r\big(\boldsymbol{x}_\mathrm{r}(t)\big).$$

Proceeding in a similar way as before yields the following state-independent NLSE:

$$\boldsymbol{0} = \boldsymbol{f}\big(\boldsymbol{v}_{ik}\,x_{\mathrm{r},i}(t_k^*),\, r(x_{\mathrm{r},i}(t_k^*))\big) - \boldsymbol{v}_{ik}\,s_v\big(x_{\mathrm{r},i}(t_k^*)\big).$$

This delivers a feasible and numerically efficient nonlinear moment matching algorithm.

Chair of Automatic Control
Prof. Dr.-Ing. habil. Boris Lohmann

www.rt.mw.tum.de
{maria.cruz, raphael.gebhart}@tum.de