Control synthesis of homogeneous approximations of nonlinear systems

Marcin Korzeń[0000−0002−1306−7146], Jarosław Woźniak[0000−0001−5930−0890], Grigory Sklyar[0000−0003−4588−9926], and Mateusz Firkowski[0000−0002−4246−9523]

West Pomeranian University of Technology in Szczecin,
Faculty of Computer Science and Information Technology,
Żołnierska str. 49, 71-210 Szczecin, Poland
http://www.wi.zut.edu.pl/
{mkorzen,jwozniak,gsklyar,mfirkowski}@zut.edu.pl

Abstract. The objective of the paper is to describe computational methods of control synthesis for a certain class of nonlinear driftless control systems. Such systems are previously found to be simplifications (called homogeneous approximations) of more complicated nonlinear systems that still preserve most crucial properties of the original ones like controllability. The class of systems in question have a special feed-forward form that is sufficiently easy to integrate and allows to solve concrete problems in control theory. Here we continue our research with describing the computational procedure for control synthesis as the extension of existing software libraries in Python language. We show that our approach leads to faster computation times compared to standard methods. The results are illustrated with some numerical experiments and simulations.

Keywords: control synthesis · nonlinear system · nonlinear approximation · homogeneous approximation · computational procedures.

1 Introduction

In this paper we continue our investigation from [11], namely we consider control systems – nonlinear with respect to state and linear with respect to controls – namely (driftless) systems of the form

$$\dot{x} = \sum_{i=0}^{m} X_i(x) u_i, \tag{1}$$

where $X_i(x)$ are real analytic vector fields in the neighborhood of the origin in $\mathbb{R}^{N+1}$. It is one of class of systems widely considered in modern control theory.

Control theory is an interdisciplinary science that combines analysis and mathematical modeling of systems treated as dynamical systems with control. Most often, a dynamical system is described using different types of differential equations, e.g. linear/nonlinear or ordinary/partial. One of the most important

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

property of control theory is controllability, which describes possibility of finding the external input that carries out the system in a finite period of time to a given state under the fulfillment of initial conditions.

The intensive development of modern control theory led to the introduction of new methods that allowed to analyze the properties of systems described by nonlinear differential equations. The classical approaches based on linearization and the other one on concepts of differential geometry (Lie brackets theory) (cf. [18] on both approaches). Linearization, i.e. replacing a nonlinear system with a linear one, although being useful in many cases, may result in the loss of structural properties of the original system and therefore cannot be freely applied. It means that in order to approximate complex nonlinear systems we should use simpler nonlinear systems. One of possible ways of choosing those simpler systems is a homogeneous approximation method and was extensively developed over the last four decades [1,2,4,6,9,14,15]. This lead to the systems that have a special, feed-forward form, that allows to consider new approaches to their further analysis. In this paper we will present some new numerical methods of control synthesis for such systems. It is worth to notice that all homogeneous systems can be described in the feed-forward form, but not the other way around. Thus the methods presented below can be applied in general for a larger class of systems, that is all feed-forward systems.

2 Numerical control synthesis in details

Control system. Here we pick up the investigations from [11] and consider homogeneous approximation of system (1) with two controls, u_0 and u_1, and we assume that it is already given in the feed-forward form,

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1(x_0) \\ f_1(x_0, x_1) \\ \vdots \\ f_N(x_0, \ldots, x_{N-1}) \end{pmatrix} u_0 + \begin{pmatrix} g_0 \\ g_1(x_0) \\ g_1(x_0, x_1) \\ \vdots \\ g_N(x_0, \ldots, x_{N-1}) \end{pmatrix} u_1 \tag{2}$$

that is f_k and g_k depend only on $x_0, \ldots, x_{k-1}$ for all $k = 1, \ldots, N$, and f_0, g_0 are constants. For such a system, with given controls u_j $(j = 0, 1)$ we can describe its behavior using only a successive integration of consecutive differential equations. Direct feed-forward integration can be troublesome because of nested integrations and some kind of interpolation of intermediate results would be helpful. Thus, a proper numerical representation for control signal and state variables is an important issue in numerical practice.

Numerical representation. For this tasks we can use representations based on Chebyshev polynomials. Chebyshev interpolation poses a very robust and convenient framework for a different types of computations with continuous functions [3,7,10,12]. We can use this framework for solving differential equations in

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

a feed-forward form both for the integration and for storing intermediate results. If the functions f_k & g_k and controls u_j are polynomials, then using Chebyshev interpolation we can obtain fully numerically accurate solutions. To implement such solutions we only need operations of integration, multiplication and addition of Chebyshev interpolators (or equivalently Chebyshev polynomials).

We assume that the control signal is a polynomial of degree n written in the Chebyshev basis of the first kind T_j $(j = 0, \ldots, n)$, that is

$$f_n(t) = \frac{1}{2}a_0 + \sum_{j=1}^{n} a_j T_j(t).$$

We will not use this form directly, but the barycentric interpolation on the Chebyshev points of the second kind (or the Chebyshev extreme points) instead [3].

The difference between Chebyshev interpolator and Chebyshev polynomial approximation is that in the first case we represent functions by the sequence of function values in Chebyshev nodes and in the second case we represent functions via sequence of coefficients in Chebyshev polynomials base. Both cases are equivalent and one can easily change the representation from one form to another, if needed. For example, for evaluating functions represented in Chebyshev nodes we have a nice barycentric interpolation procedure [3], the analogue for a polynomial representation is Clenshaw algorithm [5].

Both representations are widely used in practice, e.g. packages Chebfun [7] and PaCal [10] are based on barycentric interpolation, while package polynomial.chebyshev from numpy provides common operations on Chebyshev polynomial expansion. The transformation between both representations (called Chebyshev transformation), i.e. transition from the interpolation using Chebyshev nodes to coefficients of expansion in the Chebyshev basis, can be effectively performed using Fast Fourier Transform [12, ch. 4].

Representation using interpolation is very convenient for most algebraic operations on such interpolators and function evaluation via barycentric procedure. However, the cumulative integration is more effective when performed using Chebyshev expansion [12, ch. 2], as we have a direct formula

$$\int_0^t f_n(\tau)d\tau = \frac{1}{2}c_0 + \sum_{j=1}^{n+1} c_j T_j(t),$$

where constant c_0 is determined from the initial condition, and

$$c_j = \frac{a_{j-1} - a_{j+1}}{2j}, \text{ for } j \geq 1,$$

with $a_{n+1} = a_{n+2} = 0$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

Representation by Chebyshev interpolation. The interpolator Y is represented by values of functions y_j in Chebyshev nodes t_j:

$$Y = \begin{pmatrix} t_0, & t_1, & \dots, & t_n \\ y_0, & y_1, & \dots, & y_n \end{pmatrix}.$$

All operations used in system trajectory determining procedure are closed with respect to Chebyshev representation. For example, the sum of interpolators with the same set of Chebyshev nodes is given by

$$Y + Z = \begin{pmatrix} t_0, & t_1, & \dots, & t_n \\ y_0 + z_0, & y_1 + z_1, & \dots, & y_n + z_n \end{pmatrix},$$

and multiplication is given by

$$YZ = \begin{pmatrix} t_0, & t_1, & \dots, & t_{n+m} \\ Y(t_0)Z(t_0), & Y(t_1)Z(t_1), & \dots, & Y(t_{n+m})Z(t_{n+m}) \end{pmatrix},$$

where

$$Y(t) = \frac{\displaystyle\sum_{j=0}^{n} \frac{w_j}{t - t_j} y_j}{\displaystyle\sum_{j=0}^{n} \frac{w_j}{t - t_j}},$$

$t_j = \cos\dfrac{j\pi}{n}$ are Chebyshev nodes of the second kind, y_i are interpolated function values, and w_i are proper (barycentric) weights (cf. [3]) defined as the sequence

$$w = \left(\frac{1}{2}, -1, 1, -1, \dots, (-1)^{n-2}, (-1)^{n-1}, \frac{(-1)^n}{2}\right).$$

Recalling that the class of analyzed systems has a special feed-forward form, we can easily approach it directly using a presented Chebyshev framework. Computations with this representation are fast and numerically very stable, accuracy is typically close to the machine precision. When we restrict control signals u_i and functions f_k & g_k to polynomials then the proposed procedure will be analytically exact.

Representation by piecewise functions. Chebyshev expansion is valid only for regular continuous functions on closed interval (see e.g. [12]), but we can extend such framework to continuous piecewise functions, where each segment is represented by a Chebyshev interpolator. This is a natural extension and to perform algebraic operations we should find a common nested set of intervals. This representation is a bit more complicated as we have to aggregate computations from each sub-interval.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

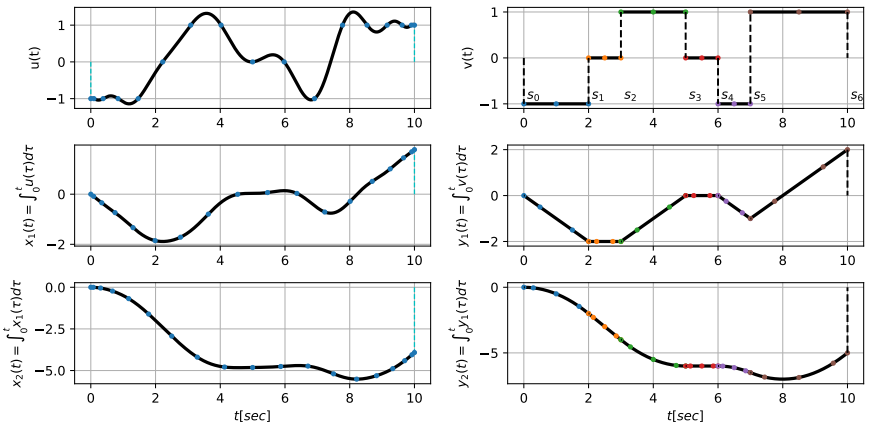

Fig. 1. Illustration of the proposed integration procedure using Chebyshev interpolation (left side) and piecewise Chebyshev interpolation (right side) for the system $\dot{x}_1 = u(t)$, $\dot{x}_2 = x_1$. Interpolation nodes are presented as dots, s_ℓ denote switching points for the case of the bang-bang control.

Note that piecewise constant functions play an important role in the control theory, and they are the main tool in the optimal control synthesis investigations [18]. In the case of bang-bang controls (piecewise constant functions) the resulting trajectories will consist also of piecewise, but not constant functions in general. Figure 1 presents comparison of two considered representations and results of simple preliminary computations. Notice that each integration step increases degrees of polynomials and, in consequence, adds some additional nodes to the interpolators.

Numerical synthesis of the controls. The control synthesis task is to find such control signals u_i, that would take the system for a given initial point $X(0)$ to the origin $\mathcal{O}$. Both considered types of representations provide two different ways to produce control signals using polynomial control and bang-bang control. We consider two optimization procedures for bang-bang control.

Details of all three considered approaches are as follows.

1. **Chebyshev or polynomial control**: In this case we represent control signals via Chebyshev barycentric interpolation, and system coordinates are represented also by Chebyshev barycentric interpolation on interval $[0, T_f]$, where time of system evolution, T_f, is given up front. To find the controls we use the optimization procedure with goal function Q_P of the form

$$Q_P(U_0, U_1) = \|x_f\|_2^2 + \alpha \sum_{k=0}^{n} \|x_k\|_2^2$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

where

$$U_0 = \begin{pmatrix} t_1, & t_2, & \ldots, & t_n \\ u_{01}, & u_{02}, & \ldots, & u_{0n} \end{pmatrix}, \quad U_1 = \begin{pmatrix} t_1, & t_2, & \ldots, & t_n \\ u_{11}, & u_{12}, & \ldots, & u_{1n} \end{pmatrix},$$

u_{ij} are values of control signals u_i $(i = 0, 1)$ in Chebyshev nodes t_j $(j = 0, 1, \ldots, n)$ scaled to the interval $[0, T_f]$, $\alpha > 0$ is a (small) regularization parameter, $\|x_f\|_2^2 = \sum_{k=0}^{n} |x_k(T_f)|^2$, $\|x_k\|_2^2 = \int_0^{T_f} |x_k(t)|^2 dt$. As final time T_f is fixed, Chebyshev nodes t_j are also fixed, and the cost function depends on interpolated nodes values u_{ij} only. As the optimization procedure we use here the l-bfgs-b procedure from scipy [13], with additional bounds on maximal values of nodes of the control signals.

2. **Bang-bang control 0**: Here we represent the control signals using piecewise constant functions on interval $[0, T_f]$ (with fixed T_f) and switching points $0 \le s_1 < \ldots < s_p \le T_f$, while system coordinates are represented by piecewise Chebyshev interpolation. In this case the cost function Q_0 is similar to the previous case, namely

$$Q_0(U_0, U_1) = \|x_f\|_2^2 + \alpha \sum_{k=0}^{n} \|x_k\|_2^2$$

where

$$U_i(t) = \begin{cases} 1, & \text{for } t \in [s_{4\ell}, s_{4\ell+1}) \\ 0, & \text{for } t \in [s_{4\ell+1}, s_{4\ell+2}) \\ -1, & \text{for } t \in [s_{4\ell+2}, s_{4\ell+3}) \\ 0, & \text{for } t \in [s_{4\ell+3}, s_{4\ell+4}), \end{cases}$$

$s_0 = 0$, $s_{p+1} = T_f$, with constrains

$$\begin{aligned} & s_1 \ge 0, \\ & s_\ell \le s_{\ell+1} \, (\ell = 0, 1, \ldots, p-1), \\ & s_p \le s_{p+1} = T_f, \end{aligned}$$

but in this case the cost function depends on switching points s_j only (see Fig. 1). Both control functions U_i are optimized using independent sequences of switching points. As we have linear constraints here this time the optimization procedure we use is trust region method, namely scipy's trust-constr procedure.

3. **Bang-bang control 1**: After careful investigation of the above optimization problem we observed that we can simplify the procedure by using interval lengths instead of switching points as variables in the procedure. In this case we are able to drop fully linear constraints and use the bound-type constraints. Here again we represent the control signals using piecewise constant functions, but on unknown at first interval $[0, T_f]$, and by widths of

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

interpolation intervals $\Delta s_\ell := s_\ell - s_{\ell-1} \geq 0$, $\ell = 1, 2 \ldots, p+1$ (instead switching points), and system coordinates are again represented by piecewise Chebyshev interpolation. As the optimization procedure we use the same procedures – the l-bfgs-b procedure from scipy [13] – with bounds on maximal values of nodes of the control signals, but this time we are able to optimize the time T_f of control of the system, if possible, by including an additional term to the cost function. Thus, the new cost function Q_1 takes the form

$$Q_1(U_0, U_1) = \|x_f\|_2^2 + \alpha \sum_{k=0}^{n} \|x_k\|_2^2 + \beta \sum_{\ell=1}^{p} \Delta s_\ell,$$

where $\alpha > 0$ and $\beta > 0$ are (small) regularization parameters,

$$U_i(t) = \begin{cases} 1, & \text{for } t \in [s_{4\ell}, s_{4\ell+1}) \\ 0, & \text{for } t \in [s_{4\ell+1}, s_{4\ell+2}) \\ -1, & \text{for } t \in [s_{4\ell+2}, s_{4\ell+3}) \\ 0, & \text{for } t \in [s_{4\ell+3}, s_{4\ell+4}), \end{cases}$$

and $s_\ell = \sum_{i=1}^{\ell} \Delta s_i$, with constrains

$$\Delta s_\ell \geq 0 \, (\ell = 1, 2 \ldots, p+1).$$

As the reference procedure for all three cases we will use a general purpose ode solver odeint function from scipy. Experimental part was prepared using Python computational environment with numpy [8] libraries mainly for arrays, scipy [17] for optimization and ODE solvers.

3 Numerical experiments

Systems. In the experimental part we consider three control systems: one based on real-life model, *sys0*, of the form

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u_0 + \begin{pmatrix} 0 \\ 1 \\ x_0 \\ -\frac{x_0^2}{2} \\ \frac{x_0^3}{6} \end{pmatrix} u_1, \tag{3}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

which is the homogeneous approximation of the truck with trailers system (see [11]), and two artificially generated ones, namely *sys1*, that is

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ x_1 \\ 0 \\ \frac{x_1 x_2}{6} \end{pmatrix} u_0 + \begin{pmatrix} 0 \\ x_0 \\ 0 \\ -\frac{x_2^2}{4} \\ 0 \end{pmatrix} u_1, \tag{4}$$

and *sys2*, given by

$$\begin{pmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 1 \\ -x_0 x_1 \\ -\frac{x_0 x_1^2}{2} \\ -\frac{x_0 x_1^3}{6} \end{pmatrix} u_2. \tag{5}$$

Results. We use two quality measures in all cases, that is ℓ^2-norm of differences between the obtained final position and the goal (here the origin $\mathcal{O}$),

$$\|x_f\|_2 = \sqrt{\sum_{k=0}^{4} |x_k(T_f)|^2}, \tag{6}$$

and ℓ^1-norm of those differences,

$$\|x_f\|_1 = \sum_{k=0}^{4} |x_k(T_f)|, \tag{7}$$

but in the case of piece-wise algorithm 1 (bang-bang control 1) we also compare the time of systems' evolution, T_f, against other algorithms, where we used the same fixed value of $T_f = 8$. In all cases we check computation times, namely we find time of establishing the control signals (t_{cont}) and time of constructing the trajectories using our feed-forward methods (t_{f-f}) and compare it to the reference time (t_{ref}) of trajectory constructions with the general purpose ode solver odeint function from scipy. Detailed results are summarized in Table 1, figures 2, 3 and 4 present sample trajectories for different initial positions.

Concerning computational time, one can observe that piece-wise algorithm 1 (bang-bang 1 controls) was able to cut the systems' evolution time, in two cases halving it, which is very promising for the applications for controllability related problems for original, more complicated nonlinear systems (cf. [11, 16]). In two cases (*'sys0'* and *'sys1'*) control synthesis computation times of all algorithms were comparable, but for system *'sys2'* Chebyshev algorithm was significantly faster. Direct comparison of our methods (Chebyshev and piecewise functions) is not simple as the number of nodes influences very differently the computational complexity in both methods. For example, for five switching points we

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

Table 1. Comparison of results of control synthesis experiments, where $|x_0|_2$ denotes ℓ^2 norm of initial state of the system,$|x_f|_2$ – ℓ^2 norm of final state of the system, $|x_f|_1$ – ℓ^1 norm of final state of the system, T_f – total time of system evolution, t_{cont} – computation time of control synthesis, t_{f-f} – computation time of finding the trajectory using our feed-forward approach, t_{ref} – computation time of finding the trajectory using reference procedures, ie. function odeint from scipy.

system	method	nodes	$\|x_0\|_2$	$\|x_f\|_2$	$\|x_f\|_1$	T_f	t_{cont}	t_{f-f}	t_{ref}
sys0	cheb	5	1.11	**0.00155**	0.00220	8	1.46	0.00320	0.304
		10	1.11	0.00214	**0.00155**	8	6.51	0.00385	0.357
		20	1.11	0.00354	0.00217	8	13.2	0.00395	0.329
		50	1.11	0.00814	0.00399	8	41.6	0.00383	0.338
	piece0	5	1.11	0.01030	0.00769	8	23.8	0.0229	0.468
		7	1.11	0.01020	0.00595	8	39.2	0.0280	0.393
		9	1.11	0.00884	0.00379	8	59.1	0.0318	0.423
	piece1	5	1.11	0.00740	0.00594	4.14	14.2	0.0157	0.170
		7	1.11	0.00610	0.00384	3.77	39.4	0.0206	0.146
		9	1.11	0.00759	0.00330	**3.61**	50.1	0.0234	0.162
sys1	cheb	5	1.11	0.00364	0.01630	8	1.92	0.00350	0.318
		10	1.11	0.00368	0.01380	8	5.93	0.00402	0.343
		20	1.11	**0.00266**	0.00595	8	18.6	0.00400	0.284
		50	1.11	0.0103	0.01020	8	45.8	0.00409	0.273
	piece0	5	1.11	0.0314	0.01710	8	29.2	0.0282	0.467
		7	1.11	0.0244	0.01060	8	48.1	0.0330	0.436
		9	1.11	0.0131	0.01160	8	74.5	0.0405	0.413
	piece1	5	1.11	0.0135	0.00827	5	30.3	0.0256	0.276
		7	1.11	0.00738	**0.00274**	**3.99**	41.6	0.0270	0.174
		9	1.11	0.00724	0.00335	4.56	63.0	0.0308	0.198
sys2	cheb	5	2.19	0.0542	0.0191	8	2.25	0.00411	0.342
		10	2.19	0.0529	0.0232	8	5.83	0.00479	0.284
		20	2.19	0.0534	0.0188	8	15.4	0.00551	0.300
		50	2.19	0.0579	0.0217	8	68.8	0.00642	0.373
	piece0	5	2.19	0.0600	0.0318	8	33.4	0.0339	0.308
		7	2.19	0.0536	0.0286	8	72.0	0.0526	0.427
		9	2.19	0.0532	0.0243	8	135	0.0713	0.573
	piece1	5	2.19	0.0460	0.0222	**6.60**	31.9	0.0317	0.263
		7	2.19	0.0294	0.0210	7.91	105	0.0472	0.357
		9	2.19	**0.0269**	**0.0111**	7.36	168	0.0672	0.545

obtain five interpolators and 15 interpolating nodes (three per each point), and calculating values of the function requires more numerical steps than calculating the analogous value for Chebyshev polynomial of 15 degree. Experiments show significant time advantage over piecewise algorithms for the same number nodes used, especially for trajectory computation times. Time needed for syn-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

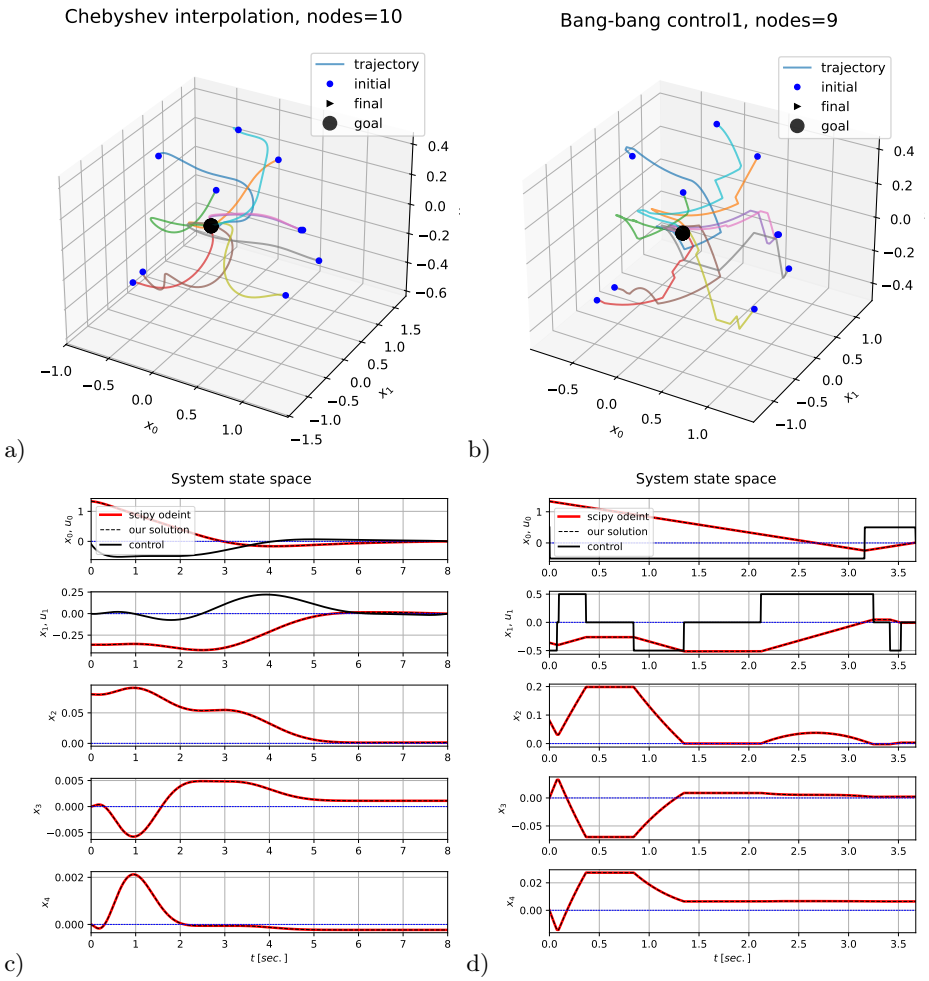

Fig. 2. Illustration of the proposed approach with the system *'sys0'* (3); a,b) projection of system's trajectories to 3D subspaces starting from different initial points. c,d) exemplary solution: control signals u_0, u_1 (solid black), black dashed lines represent solution in our representation – system's state space in separate coordinates x_k, $i = 0, 1, 2, 3, 4$, red solid lines – the same solution given by scipi.odeint.

thesis control does not show such large differences, most probably due to the fact that the goal function Q_P has an easier form but has to be called more times in the optimization process. Experiments show that those synthesis times are comparable in the cases of 50 Chebyshev nodes and only 7 piecewise nodes (switching points). In all 30 experiments computational times of generating systems' trajectories were much faster using our new method than with the general

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

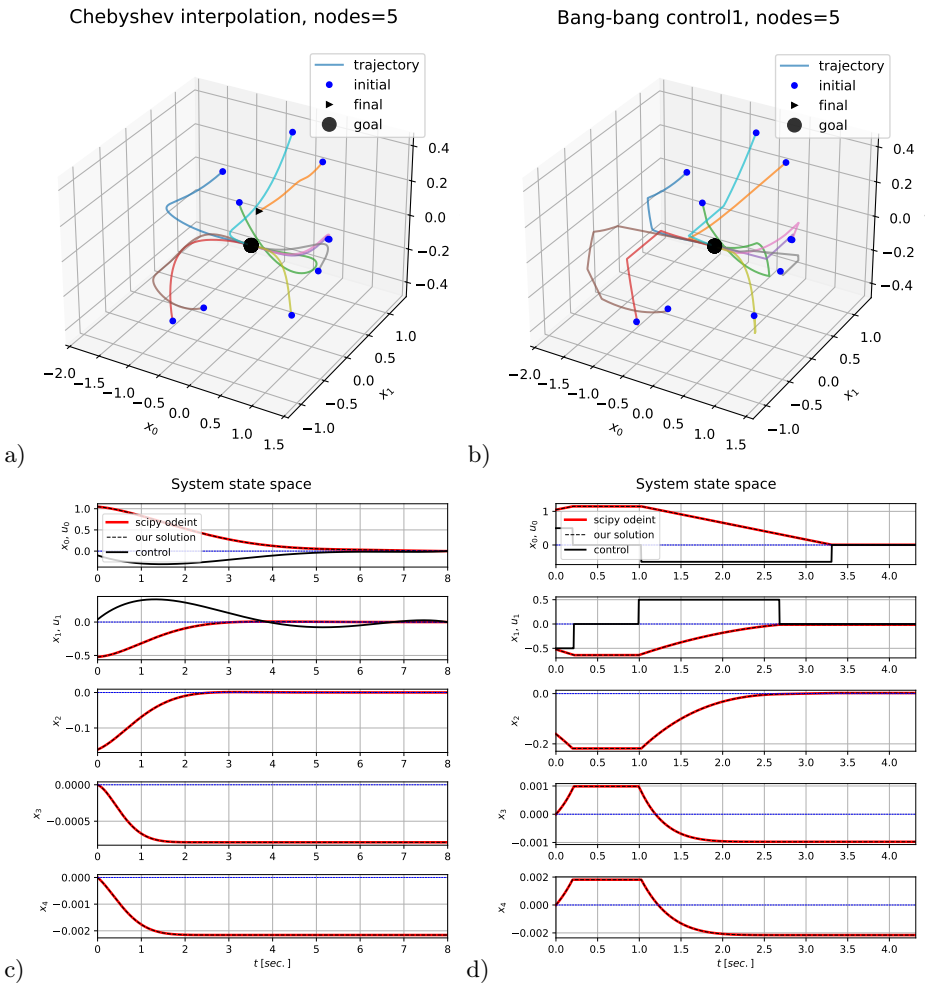

Fig. 3. Illustration of the proposed approach with the system *'sys1'* (4); a,b) projection of system's trajectories to 3D subspaces starting from different initial points. c,d) exemplary solution: control signals u_0, u_1 (solid black), black dashed lines represent solution in our representation – system's state space in separate coordinates x_k, $i = 0, 1, 2, 3, 4$, red solid lines – the same solution given by scipi.odeint.

reference procedures (function odeint from scipy), being from ten times to even one hundred times faster.

Concerning quality of the goal objectives – reaching the origin $\mathcal{O}$ – one can observe that in all cases our trajectories tended in the direction of the goal. All goal reaching measures give comparable results between Chebyshev and piece-wise methods, but in general piece-wise algorithm 1 (bang-bang 1 controls) seems to give slightly better results than piece-wise algorithm 0 (bang-bang 0 controls).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

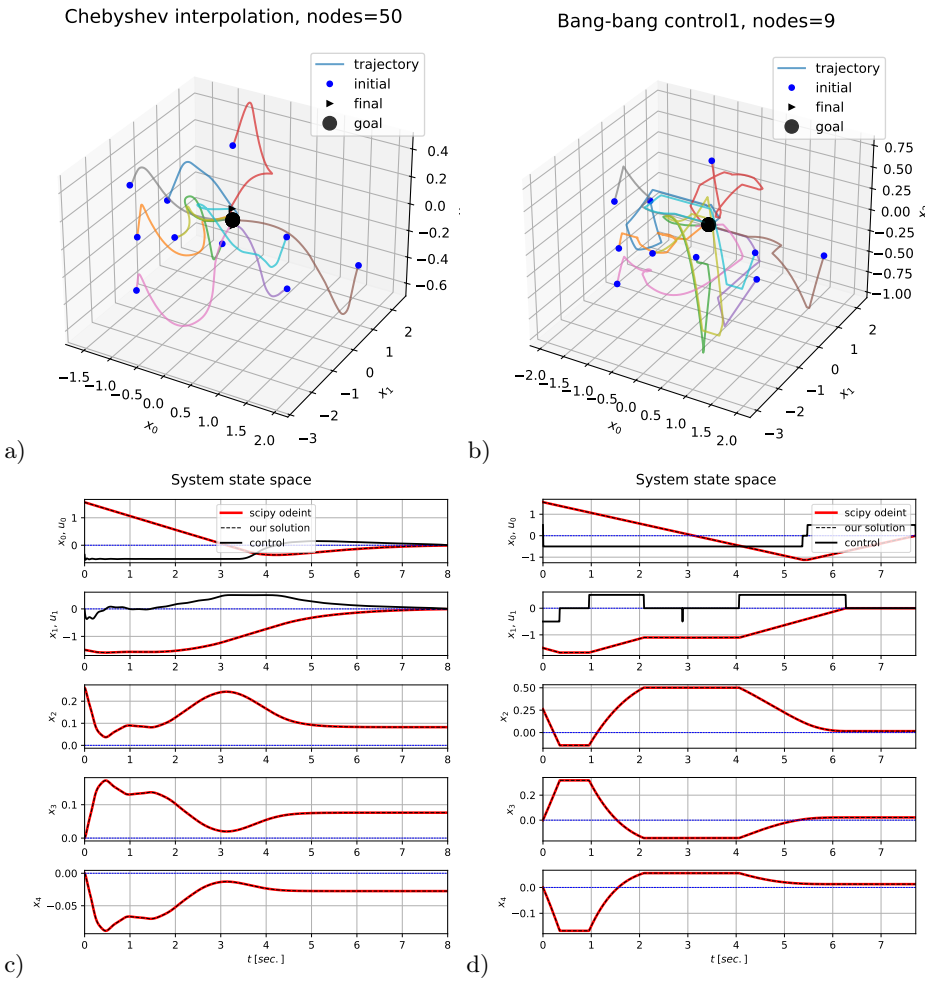

Fig. 4. Illustration of the proposed approach with the system *'sys2'* (5); a,b) projection of system's trajectories to 3D subspaces starting from different initial points. c,d) exemplary solution: control signals u_0, u_1 (solid black), black dashed lines represent solution in our representation – system's state space in separate coordinates x_k, $i = 0, 1, 2, 3, 4$, red solid lines – the same solution given by scipi.odeint.

4 Summary

In this paper we presented the procedure of determining controls for homogeneous nonlinear systems from a computational point of view, and we provided the numerical experiments with some selected nonlinear control systems in the feed-forward form, that could arise from homogeneous approximations. After comparing the system trajectories, we briefly discussed the quality of such controls. The experiments confirmed that the theoretical results concerning homo-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

geneous approximations of nonlinear systems can be used in practice, and the need to construct suitable software libraries to be used in possible applications – e.g. in practical control design – is evident.

The approach presented in this paper can be further investigated, for example trying to minimize not only time, but also energy norm of control signals.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agrachev, A., Marigo, A.: Nonholonomic tangent spaces: intrinsic construction and rigid dimensions. Electron. Res. Announc. Amer. Math. Soc. **9**, 111–120 (2003)
2. Bellaïche, A.: The tangent space in sub-Riemannian geometry. Progress in Mathematics **144**, 4–78 (1996)
3. Berrut, J.P., Trefethen, L.N.: Barycentric lagrange interpolation. SIAM Review **46**(3), 501–517 (2004)
4. Bianchini, R., Stefani, G.: Graded approximation and controllability along a trajectory. SIAM J. Control Optimiz. **28**, 903–924 (1990)
5. Clenshaw, C.W.: A note on the summation of Chebyshev series. Mathematical Tables and other Aids to Computation **9**, 118–120 (1955)
6. Crouch, P.E.: Solvable approximations to control systems. SIAM J. Control Optimiz. **22**, 40–54 (1984)
7. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun Guide. Pafnuty Publications (2014), http://www.chebfun.org/docs/guide/
8. Harris, C.R., et al.: Array programming with NumPy. Nature **585**(7825), 357–362 (Sep 2020)
9. Hermes, H.: Nilpotent and high-order approximations of vector field systems. SIAM Rev. **33**, 238–264 (1991)
10. Jaroszewicz, S., Korzeń, M.: Arithmetic Operations on Independent Random Variables: A Numerical Approach. SIAM J. Sci. Comp. **34**(4), A1241–A1265 (2012)
11. Korzeń, M., Sklyar, G.M., Ignatovich, S.Y., Woźniak, J.: Computational aspects of homogeneous approximations of nonlinear systems. Proc. ICCS 2024 pp. 368–382 (2024)
12. Mason, J., Handscomb, D.: Chebyshev Polynomials. CRC Press (2002)
13. Morales, J.L., Nocedal, J.: Remark on "algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization". ACM Trans. Math. Softw. **38**(1) (Dec 2011)
14. Sklyar, G.M., Ignatovich, S.Y.: Approximation of time-optimal control problems via nonlinear power moment min-problems. SIAM J. Control Optimiz. **42**, 1325–1346 (2003)
15. Sklyar, G.M., Ignatovich, S.Y.: Description of all privileged coordinates in the homogeneous approximation problem for nonlinear control systems. C. R. Math. Acad. Sci. Paris **344**, 109–114 (2007)
16. Sklyar, G.M., Ignatovich, S.Y.: Free algebras and noncommutative power series in the analysis of nonlinear control systems: an application to approximation problems. Dissertationes Math. (Rozprawy Mat.) (2014), 1–88. (2014)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1

17. Virtanen, P., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
18. Zabczyk, J.: Mathematical Control Theory: An Introduction. Birkhäuser, Basel (2008)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_1