Motion planning for the 2D Stokes equations

Thomas Meurer and Mourad Saidani

Abstract—A flatness–based design systematics is presented to realize desired velocity profiles for the Stokes equations in a 2D domain and spatially distributed interior control. The approach combines early and late lumping techniques and utilizes the spectral properties of the system operator. Re–summation techniques are integrated to enhance the domain of applicability as is illustrated in a simulation scenario.

I. INTRODUCTION

In the last decades the analysis and the optimization of fluid dynamical systems has gained increasing interest due to the availability of sophisticated numerical algorithms and the progress in high–performance computing. Moreover, different approaches for the active control of fluid flow have been developed utilizing, e.g., linear, nonlinear, and optimal control methods to achieve transition control from laminar to turbulent, separation control, drag or noise reduction, or mixing enhancement (see, e.g., [1], [2], [3], [4], [5], [6], [7], [8] and the many references therein).

In the following, the so–called motion planning problem is considered for the Stokes equations in a bounded rectangular 2D spatial domain with the control located in the domain’s interior. Motion planning herein refers to the determination of open–loop input trajectories to realize a desired spatial–temporal velocity and pressure distribution. Recently developed flatness–based techniques for distributed–parameter control problems with Riesz spectral system operator in [9], [10] build the center of our design approach. Flatness thereof refers to the determination of a flat or basic output, which enables a (differential) parametrization of the state and input variables, see, e.g., [11], [12], [13], [14], [15] and the references therein. In the context of flow control, related techniques are presented in [16] for the linearized Navier–Stokes equations with boundary control in a double infinite–dimensional domain. Thereby, the Fourier transformation is applied to consider the transformed equations in a wave number space. However, this approach is not applicable to both bounded domains and spatially distributed in–domain control as is considered in this contribution.

For this, early and late lumping concepts are combined in an efficient semi–numerical flatness–based design technique for Stokes equations. Suitable semi–discretization of the governing partial differential equations resulting in a linear differential–algebraic system and modal reduction are utilized to determine the spectral or modal system representation. Based on this formulation a flat output is systematically constructed to achieve a differential parametrization of the spectral states and the input variables. The convergence of the parametrization is addressed both for the finite–dimensional approximation and in the limit as the discretization approaches the continuous formulation. In the latter case, the parametrization yields differential operators of infinite order, which correspond to so–called Weierstrass canonical products in the Laplace domain. Their convergence essentially relies on the distribution of the eigenvalues. Similar to the results in [17] for a wave equation and [10] for a diffusion–reaction equation it is shown for the 2D case that the parametrizations in general diverge. In order to overcome this, re–summation techniques are integrated into the design, which enable to recover meaningful input trajectories also under divergent conditions [13], [15]. With this, the assignment of appropriate desired trajectories for the basic output is considered in Gevrey classes to realize finite time transitions between stationary velocity and pressure profiles. Differing from optimal control techniques for flow problems the approach yields a one–to–one correspondence between system trajectories and hence immediately accounts for simple re–planning strategies as is illustrated in a simulation scenario. However, steady–state profiles satisfying certain optimality conditions can be readily included into the presented flatness–based design systematics.

The paper is organized as follows. The Stokes system with in–domain control and its semi–discrete approximation by making use of the so–called MAC scheme are introduced in Section II. Flatness–based motion planning and feedforward control is considered in Section III by systematically constructing a flat output for the finite–dimensional approximation to differentially parametrize states and inputs. Convergence of the parametrization is studied in the limit as the semi–discrete system approaches the continuous formulation. Simulation results are presented in Section IV. Some final remarks in Section V conclude the paper.

II. PROBLEM FORMULATION

In the following, the Stokes system is considered in dimensionless form governed by

\[ \frac{\partial}{\partial t}v(z, t) - \frac{1}{Re} \Delta v(z, t) + \nabla p(z, t) = \sum_{l=1}^{m} b_l(z) u_l(t) \quad (1a) \]

\[ \nabla \cdot v(z, t) = 0 \quad (1b) \]

for \((z, t) \in \Omega \times R^+\) with \(\Omega = (-1, 1) \times (-1, 1)\). Eqns. (1a) and (1b) can be directly deduced from the Navier–Stokes equations for small Reynolds numbers Re. Herein, \(v(z, t) = [v_1(z, t), v_2(z, t)]^T\) is the vector of velocities in the \(z_1\)– and \(z_2\)–directions, \(p(z, t)\) is the pressure distribution, \(b_l(z) =\)
More general boundary conditions can be similarly addressed. The system is initially assumed to be in steady state

\[
v(z, t) = 0, \quad z \in \partial \Omega, \quad t > 0.
\]  

(1c)

Boundary conditions are imposed by suitably evaluating (1c) along the boundary cells. For the numerical solution, it is possible to eliminate the implicit dependency on \( p(z, t) \) by observing that \( \nabla \cdot (1a) \) in view of (1b) results in the Poisson equation \( \Delta p(z, t) = \sum_{i=1}^{m} \nabla b_i(z) u_i(t) \), which can be similarly discretized on the introduced grid. However, to avoid the differential–algebraic system structure a penalization approach is considered, see, e.g. [20], [21]. For this, introduce the vectors \( x(t) \) and \( p(t) \) summarizing the values of \( v^i_j(t) \), \( k = 1, 2 \), and \( p^{i,j}(t) \) at the respective (staggered) grid nodes. The MAC semi–discretization can be hence reformulated according to

\[
\partial_t x(t) = A_1 x(t) + A_2 p(t) + B_1 u(t)
\]  

(2a)

\[
\epsilon p(t) = A_2 x(t)
\]  

(2b)

Herein, \( 0 < \epsilon \ll 1 \) denotes the penalization parameter, \( A_1 x(t) \). \( A_1 p(t) \) as well as \( A_2 x(t) \) correspond to the discretization of \( 1/\Re \Delta v(z, t), \nabla p(z, t), \) and \( \nabla \cdot v(z, t) \), respectively, and \( B_1 u(t) \) summarizes the actuation with the vector of control inputs \( u(t) = [u_1(t), \ldots, u_m(t)]^T \). The proper selection of \( \epsilon \) relies on a trade–off between accuracy and numerical ill–conditioning. Subsequently, \( \epsilon = -2 \Re / c \) with \( c = 10^7 \) is used as recommended in [20, Sec. 2.9] for 64–bit floating point arithmetic.

### B. Spectral analysis and modal representation

The finite–dimensional approximation (2) enables an efficient numerical analysis of the spectral properties of the Stokes system (1). These are exploited in the subsequent sections to solve the motion planning problem. In particular, substituting (2b) into (2a) yields

\[
\partial_t x(t) = Ax(t) + B_1 u(t), \quad A := A_1 + \frac{1}{\epsilon} A_2 A_2
\]  

with \( A = A^T \). Let subsequently \( \lambda \) and \( \phi \) denote eigenvalues and right–eigenvectors of \( A \), i.e.

\[
A \phi = \lambda \phi.
\]  

(3)

Then applying the regular space–translation

\[
x(t) = V \eta(t)
\]  

(4a)

to (2) with the \( \dim x \times \dim x \)–matrix

\[
V = \left[ \phi_1, \ldots, \phi_{\dim x} \right]
\]  

(4b)

satisfying \( AV = V \Lambda \), where \( \Lambda = \text{diag}(\lambda_k)_{k=1, \ldots, \dim x} \) for \( 0 > \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\dim x} \), yields the modal system representation

\[
\partial_t \eta(t) = \Lambda \eta(t) + B u(t)
\]  

(5)

1The discretized original system is recovered for \( \epsilon = 0 \) in differential–algebraic form with eigenvalues at infinity. It is in particular their approximation that causes numerical difficulties. Given a non–zero \( \epsilon \ll 1 \) yields a separation of the eigenvalues of the differential subsystem from those of the algebraic subsystem, which are now finite but of several magnitudes larger. With this, standard numerical algorithms can be applied contrary to specialized techniques as considered, e.g., in [22]. These, however, can be similarly integrated into the determination of the spectral system representation.
with \( B = W^T B_1 \). Here, the matrix \( W \in \mathbb{R}^{\text{dim } x \times \text{dim } x} \) is assumed to satisfy \( WTV = I \) with \( I \) denoting the identity matrix and can be constructed using the left–eigenvectors of \( A \).

C. Modal reduction

The modal representation directly admits a modal reduction of the system dynamics by considering the projection onto the subspace \( \text{span}_{k=1,\ldots,n}(\phi_k) \), \( n < \text{dim } x \). Since \( x(t) \) is in general of high dimension depending on the spatial discretization the reduction step is an essential tool for control design. Hence, let subsequently \( \eta'(t) = [\eta_1(t), \ldots, \eta_n(t)]^T \), \( n < \text{dim } x \) refer to the reduced order system and \( \eta^s(t) = [\eta_{n+1}(t), \ldots, \eta_{\text{dim } x}(t)]^T \) to the residual state so that \( \eta(t) = \eta'(t) \oplus \eta^s(t) \) and thus

\[
x(t) = V^T \eta'(t) + V^s \eta^s(t)
\]

with \( V = [\phi_1, \ldots, \phi_n] \) and \( V^s = [\phi_{n+1}, \ldots, \phi_{\text{dim } x}] \). In view of (5) this yields

\[
\begin{align*}
\partial_t \eta'(t) &= \Lambda' \eta'(t) + B' u(t) \quad (7a) \\
\partial_t \eta^s(t) &= \Lambda^s \eta^s(t) + B^s u(t) \quad (7b)
\end{align*}
\]

with the system matrices \( \Lambda' := \text{diag}\{\lambda_k\}_{k=1,\ldots,n} \), \( \Lambda^s := \text{diag}\{\lambda_k\}_{k=n+1,\ldots,\text{dim } x} \) and \( B u(t) = B' u(t) \oplus B^s u(t) \).

Due to the arrangement of \( \lambda_k \), (7a) can be considered as the slow dynamics while (7b) represents the fast dynamics.

III. FLATNESS–BASED PARAMETRIZATION

In the following, flatness–based trajectory planning is considered for the Stokes equations taking into account the decomposition of the system dynamics according to (7).

Assumption 1. The geometric and algebraic multiplicities \( r^g_k \) and \( r^a_k \) of each eigenvalue \( \lambda_k \) coincide. The system (7a) is controllable, i.e. \( \text{rank}[B^r, \Lambda^r B^r, \ldots, (\Lambda^r)^{n-1} B^r] = n \).

Throughout the remainder \( n_0 \leq n \) refers to the number of distinct eigenvalues. Since \( r^g_k = r^a_k \), controllability requires that \( m \geq \max_{k=1,\ldots,n_0} r^a_k \).

A. Differential parametrization

The proposed design approach relies on the separation of the system dynamics by determining a differential parametrization of (7a) and utilizing (7b) in a corrector step to suitably adjust motion planning. To address the first part formally apply the Laplace transform to (7a), which yields after some simple algebraic manipulations

\[
\begin{align*}
\hat{\eta}'(s) &= -(I - s(\Lambda')^{-1})^{-1}(\Lambda')^{-1} B^r \hat{u}(s) \\
&= -\frac{D^\eta'(s)}{D^u(s)} (\Lambda')^{-1} B^r \hat{u}(s)
\end{align*}
\]

with \( s \) the Laplace operator, \( \hat{\eta}'(s), \hat{u}(s) \) the Laplace transforms of \( \eta'(t), u(t) \), and \( I \) the \( (n \times n) \) identity matrix. Since \( \Lambda' \) is a diagonal matrix, we have

\[
\begin{align*}
D^\eta'(s) &= \frac{\text{adj}(I - s(\Lambda')^{-1})}{\prod_{j=1}^{n_0} (1 - \frac{s}{\lambda_j})^{r^g_j-1}} \\
&= \text{diag}\left\{\frac{\text{adj}\left(\prod_{j=1, j \neq k}^{n_0} (1 - \frac{s}{\lambda_j})\right)}{k=1,\ldots,n} \right\} (8a) \\
D^u(s) &= \text{det}(I - s(\Lambda')^{-1}) = \prod_{j=1}^{n_0} (1 - \frac{s}{\lambda_j}) \\
&= \prod_{j=1}^{n_0} (1 - \frac{s}{\lambda_j})^{r^g_j-1}, \quad (8b)
\end{align*}
\]

where \( \text{adj}(\cdot) \) is the adjugate matrix and the denominator \( \prod_{j=1, j \neq k}^{n_0} (1 - s/\lambda_j)^{r^g_j-1} \) is used to cancel common terms. Note that \( D^\eta'(s) \) is an \( (n \times n) \) matrix while \( D^u(s) \) is scalar. Hence, introducing \( \hat{y}(s) = \hat{u}(s)/D^u(s) \) results in the operational state and input parametrization

\[
\begin{align*}
\hat{\eta}'(s) &= -D^\eta'(s)(\Lambda')^{-1} B^r \hat{y}(s) \\
\hat{u}(s) &= D^u(s) \hat{y}(s)
\end{align*}
\]

such that \( \hat{y}(s) \) can be called a flat or basic output. Recalling that \( s \) corresponds to time differentiation the equivalent time domain formulation reads as

\[
\begin{align*}
\eta'(t) &= -D^\eta'(\partial_t)(\Lambda')^{-1} B^r \hat{y}(t) \quad (10a) \\
u(t) &= D^u(\partial_t) \hat{y}(t), \quad (10b)
\end{align*}
\]

where \( D^\eta'(\partial_t) \) and \( D^u(\partial_t) \) have to be interpreted as differential operators of order \( n - 1 \) and \( n \), respectively. Hence, any trajectory \( y(t) \) for the basic output has to be at least \( n \)-times continuously differentiable.

Remark 1. In view of the following analysis based on entire function theory, we remark that (8) can be equivalently re–formulated in the form

\[
D^\eta'(s) = \text{diag}\left\{e^{F(s, q)}\right\} \times \prod_{j=1, j \neq k}^{n_0} (1 - \frac{s}{\lambda_j}) e^{F(\frac{s}{\lambda_j}, q)} \right\} \quad (11a)
\]

\[
D^u(s) = \prod_{j=1}^{n_0} (1 - \frac{s}{\lambda_j}) e^{F(\frac{s}{\lambda_j}, q)} \quad (11b)
\]

with \( F(\cdot, q) \) a function to be defined below. Hence, \( D^\eta'(s)/D^u(s) = \text{adj}(I - s(\Lambda')^{-1})/\text{det}(I - s(\Lambda')^{-1}) \).

B. Convergence analysis and continuous limit

It is obvious from (8) that convergence of the so far formal state and input parametrizations essentially relies on the eigenvalue distribution. To analyze convergence, we make use of entire function theory [23], [24] by

1) considering the continuous limit as \( n_0 \to \infty \) and hence

2) interpreting \( D^u(s) \) and the components of \( D^\eta'(s) \) as factorizations of entire functions in the complex domain \( s \in \mathbb{C} \).

This requires certain preliminary considerations. In particular, let \( n(r) \) denote the counting function of the sequence \( (\lambda_k)_{j=1,\ldots, r^g_j} \in \mathbb{N} \) defined by \( n(r) = \#\{\lambda_k, k \in \mathbb{N} : |\lambda_k| \leq r\} \), where \# refers to the number of elements in the set. Moreover, the (Weyl) asymptotic behavior of the
eigenvalues is known, see, e.g., [25, Prop. 4.14f], and yields for the 2D case
\[ \lambda_k \sim \frac{4\pi^2}{\omega_2 |\Omega|^k} \Rightarrow n(\lambda) = \omega_2 |\Omega|^{\frac{1}{2}} \lambda \]  
with \( \omega_2 = \pi \) the area covered by the unit circle. This result enables to evaluate the convergence exponent \( \nu \) (cf. Def. A.1), which plays a crucial role to determine the order of growth of an entire function. Given a sequence with countable index set, then \( \nu \) equals the order of its counting function [24, Sec. 3.2], which for \( (\lambda) \) implies
\[ \nu = \limsup_{r \to \infty} \frac{\log n(r)}{\log(r)} = 1. \]

With (12) the genus (cf. Def. A.2) of \( (\lambda) \) is \( g = 1 \). Consider now (11b) with \( n_0 \to \infty \) such that
\[ D^m(s) = \prod_{j \in \mathbb{N}} \left( 1 - \frac{s}{\lambda_j} \right) e^{F\left(\frac{s}{\lambda_j}, g\right)} \]
and let \( F(s, g) = 0 \) if \( g = 0 \) and \( F(s, g) = \sum_{i=1}^{g} s^i / i \) for \( g > 0 \) denote the so–called Weierstrass convergent functions. Then by the Hadamard factorization theorem \( D^m(s) \) is a Weierstrass canonical product of genus \( g \) and an entire function of order \( g \) equal to the convergence exponent of the its zero set \( (\lambda) \) [23]. The order thereby refers to the property that the maximal modulus \( M(r) = \max_{|s| \leq r} |D^m(s)| \) \( < \exp(r^\nu + \epsilon) \) for sufficiently large \( r \) and \( 0 < \epsilon \ll 1 \). With \( \nu = 1 \) it follows that \( g = 1 \) for the Stokes equations.

Proceeding as in [9, Prop. 2] or [10, Prop. 7] yields that \( u(t) = D^m(\partial_t) \circ y(t) \) converges uniformly for any basic output trajectory \( y(t) \in (G_{D,\alpha}(\mathbb{R}))^m \) in the Grevy class (see Def. A.3) of order \( \alpha \leq 1 / g = 1 \) provided that \( D^m(s) \) is of finite type \( \tau \), i.e. there exists a \( \tau r \leq r \) such that \( M(r) < \exp((\tau + \epsilon)r^\nu) \) for sufficiently large \( r \) and \( 0 < \epsilon \ll 1 \). Taking into account that
\[ \tau = \limsup_{r \to \infty} \frac{\log M(r)}{r} \]
it is possible to conclude in view of (13), the Weyl asymptotics (12), and \( \lambda_k < 0, k \in \mathbb{N} \) that
\[ \tau = \limsup_{r \to \infty} \frac{\log \prod_{j \in \mathbb{N}} (1 - \frac{r}{|\lambda_j|}) e^{F\left(\frac{r}{|\lambda_j|^2}, 1\right)}}{r} = \limsup_{r \to \infty} \frac{\pi}{r} \log \left( 1 + \frac{r}{|\lambda_j|} \right) \]
which implies \( \tau = \infty \). Thus, \( D^m(s) \) is of maximal type and violates the convergence conditions. Note that a similar analysis can be performed for the components of \( D^m(s) \).

As a result of the previous analysis, convergence of the formal state and input parametrizations cannot be guaranteed in the continuous limit with \( D^m(\partial_t) \) and \( D^m(\partial_i) \) interpreted as differential operators of infinite order. It has to be pointed out that this is a rather common outcome given partial differential equations with higher–dimensional domain and finite–dimensional input. Examples include the boundary controlled linear wave equation [17] or diffusion–reaction systems [10]. Nevertheless, the divergent behavior does not prevent the applicability of the formal parametrizations for motion planning. For this, suitable re–summation techniques are subsequently integrated into the design to enable convergence acceleration and the extraction of a meaningful limit given divergent series.

C. Motion planning and feedforward control

Let \( x_0^* \) and \( x_T^* \) denote desired initial and final states, then (6) yields
\[ x_i^* = V^r \eta_i^* + V^s \eta_i^{*,*}, \quad i \in \{0, T\}. \]
Neglecting the residual contributions and recalling that \( V^r \in \mathbb{R}^{\dim x \times n} \) with \( n \leq \dim x \) the pseudo–inverse can be applied to solve for
\[ \eta_i^{*,*} = (V^r)\dagger x_i^* \]
and hence to minimize \( \|x_i^* - V^r \eta_i^{*,*}\|_2^2 \). Taking into account the formal state parametrization (10a), (11a) hence implies
\[ \eta_i^{*,*} = -D^m(0) \circ (\Lambda^* - 1)^{-1} B^r y_i^* \]
Recalling that \( \Lambda^* \in \mathbb{R}^{n \times n} \) and \( B^r \in \mathbb{R}^{n \times m} \) it follows that except for \( n = m \) an over– or under–determined system of linear equations is obtained to solve for \( y_i^* \). In order to address this, let \( H = -(\Lambda^*)^{-1} B^r \) and let \( H^\dagger = (H^T H)^{-1} H^T \) denote the pseudo–inverse of \( H \). Then
\[ y_i^* = H^\dagger \eta_i^{*,*}, \quad i \in \{0, T\} \]
solves the least–squares problem
\[ \min_{y_i^*} \|\eta_i^{*,*} - H y_i^*\|_2^2 \]

Remark 2. Note that residual contributions on the resulting stationary profile \( x_i \) can be similarly addressed during motion planning. Assuming stationary conditions the input parametrization implies \( u_i = y_i^* \), which together with the stationary version of (7) provides that (14) evaluates to
\[ x_i^* = -[V^r(\Lambda^*)^{-1} B^r + V^s(\Lambda^*)^{-1} B^r] y_i^*. \]
Denoting the square bracketed term by \( H \) yields \( y_i^* = H^\dagger x_i^* \) minimizing \( \|x_i^* - H y_i^*\|_2^2 \) and taking into account the residual contributions. It should be mentioned that only a (dominant) subset of \( V^s \eta_i^{*,*} \) can be included similarly.

As a result, a desired trajectory for the basic output can be assigned in the form
\[ y_i'(t) = y_0^* + (y_T^* - y_0^*) \Phi(t). \]
with \( y_T^* \) from (15) and the function \( \Phi(t) \) satisfying \( \Phi(0) = 0 \), \( \Phi(T) = 1 \), and \( \partial_t^j \Phi(t) = 0 \) locally at \( t \in \{0, T\} \) for \( j = 1, \ldots, n \). For this, polynomials of suitable degree or Fourier series can be applied. In the following, \( \Phi(t) = g_{T, \omega}(t) \) with
\[ g_{T, \omega}(t) = \begin{cases} \frac{h_{T, \omega}(t)}{h_{T, \omega}(0)}, & t \leq 0 \\ \frac{h_{T, \omega}(t)}{h_{T, \omega}(0)}, & t \in (0, T) \\ 1, & t \geq T \end{cases} \]
is used, where \( h_{T, \omega}(t) := \exp\left( -((1 - t/T)t/T)^{-\omega} \right) \) if \( t \in (0, T) \) and \( h_{T, \omega}(t) = 0 \) else. Hence, \( \Phi(t) \in C^\infty(\mathbb{R}) \) and as
desired is locally non-analytic at \( t \in \{0, T\} \). Note also that \( \Phi(t) \) is of Gevrey order \( \alpha = 1 + 1/\omega \).

As pointed out in Section III-B, the flatness–based state and input parametrization in general diverge such that subsequently re–summation techniques are employed to recover meaningful state and input trajectories from the formal results. For this note at first that (10) admits a polynomial or series representation according to

\[
\eta^r(t) = \sum_{j=0}^{J^r} c_j^r \partial_j^r y(t), \quad u(t) = \sum_{j=0}^{J^u} c_j^u \partial_j^u y(t) \tag{18}
\]

with \( J^r = n - 1, J^u = n \) for \( D^r(s) \) and \( D^u(s) \) determined by (8) or \( J^r = J^u = \infty \) for (11). In view of (18) we apply the so–called \( (N, \xi) \)-approximate \( k \)-summation \( S^N_k \) to determine the parametrized state evolution and the feedforward control from (18), i.e.

\[
\eta^r(t) \mapsto (S^N_k \eta^r)(t), \quad u(t) \mapsto (S^N_k \eta^r)(t), \tag{19}
\]

where for \( f(t) = \sum_{j=0}^{\infty} f_j \partial_j^r y(t) \) we define

\[
(S^N_k f)(t) = \sum_{j=0}^{N} s_j(t) \frac{\xi^j}{(1+\xi)^j}, \quad s_j(t) = \sum_{i=0}^{j} f_i \partial_i^r y(t).
\]

Herein, \( N, \xi, \) and \( k \) denote degrees–of–freedom, which have to be determined appropriately. For an analysis of the properties of \( S^N_k \) in view of the solution of trajectory planning problems for distributed–parameter systems the reader is referred to [13], [10], [15].

IV. SIMULATION RESULTS

The flatness–based state and input parametrization are subsequently evaluated in a simulation scenario for Reynolds number \( Re = 1 \). As an example, the desired trajectory for the basic output is thereby chosen to realize the transition from a zero initial state \( u_0(z) = 0 \) to the final state \( u_T(z) \) corresponding to the 4–th eigenstate of the (discretized) Stokes equations within the time interval \( t \in [0, T] \) for \( T = 1/(2Re) = 1/2 \). By proceeding as in Section III-C the desired trajectory \( y^*(t) \) is assigned as (16) with \( y_0^* = 0 \) and \( y_T^* \) determined from (15) with \( \eta_{T,A}^* = 1 \) the only non–zero component of \( \eta^r_T \). The considered actuator location and the flatness–based feedforward control \( u(t) \) are shown in Figure 2 (left column). For the numerical realization an academic example is considered with \( m = 16 \) consecutively numbered patch actuators placed on the domain with each being actuated by the respective input trajectory. For their evaluation, partial summation and \( (N, \xi) \)-approximate \( k \)-summation with \( N = 70, \xi = 1, \) and \( k = 1 \) according to (19) are compared for a single input trajectory in Figure (3). The curves clearly confirm the divergent behavior as predicted in Section III-B. However, the application of the re–summation technique (19) obviously allows to extract meaningful input trajectories. Simulation results for the application of the feedforward control to the MAC discretization of Stokes equations are presented in Figure 2. Here, the flatness–based state parametrization evaluated from (19) using an \( S_1^{70,1} \) is compared to the simulation result by applying \( u^*(t) \) to the discretized system. Obviously, the desired and the obtained velocity fields match with only minor deviations. In addition, the simulation results confirm the accurate realization of the desired steady state to steady state transition within the prescribed time interval also under diverging conditions by means of suitable summation techniques.

Remark 3. The presented approach is independent of the number, shape, and orientation of the actuators on the spatial domain provided that Assumption 1 holds. Although obvious it is noted that the achievable motion planning accuracy essentially relies on a trade–off between actuator placement and number and the selection of the desired profiles.

V. CONCLUSIONS

In this contribution a flatness–based motion planning approach is presented for the Stokes system defined on a quadratic domain with interior actuation. The proposed early lumping approach is based on the modal or spectral system representation obtained from a MAC discretization and the solution of the algebraic eigenvalue problem. The modal decomposition allows to systematically determine a differential state and input parametrization in terms of a basic output. Convergence of the parametrizations is addressed by considering the limit as the discretized system approaches the continuous limit. By making use of Weyl asymptotic results for the eigenvalue distribution it is shown that the flatness–based formal parametrizations diverge in general. Nevertheless, the incorporation of appropriate re–summation techniques enables to recover meaningful results also for the divergent case as is verified in a simulation example. With this, a first extension of flatness–based methods to realize motion planning for fluid–dynamical system is achieved, which combines early and late lumping techniques for design and convergence analysis.

APPENDIX

For a thorough and detailed analysis of entire functions and their properties the reader is referred to [23], [24]. Let subsequently \( (a_n)_{n \in \mathbb{N}} \) be a sequence of (complex) numbers with \( a_n \neq 0 \) and \( \lim_{n \to \infty} |a_n| = \infty \).

Definition A.1. The infimum \( \nu \) of positive numbers \( \nu_0 \) such that \( \sum_{n=0}^{\infty} |a_n|^{-\nu_0} < \infty \) is called the convergence exponent of the sequence \( (a_n)_{n \in \mathbb{N}} \).

Definition A.2. The smallest positive integer \( \nu_0 \) for which \( \sum_{n=0}^{\infty} |a_n|^{-\nu_0} < \infty \) is denoted by \( g + 1 \) and \( g \) is called the genus of the sequence \( (a_n)_{n \in \mathbb{N}} \).

Definition A.3. A function \( f(t) \in C^\infty(\mathbb{R}) \) is in \( G_{D,\alpha}(\mathbb{R}) \), the Gevrey class of order \( \alpha \), if \( \exists D, \alpha > 0 \) such that \( \sup_{t\in\mathbb{R}} |\partial_\sigma^\alpha f(t)| \leq D^{n+1}(n!)^\alpha \) for any \( n \in \mathbb{N} \). Hence, \( f(t) \) is entire if \( \alpha < 1 \), analytic if \( \alpha = 1 \), or non–analytic if \( \alpha > 1 \).

REFERENCES

Fig. 2. Simulation results for flatness–based feedforward control using \((N, \xi)\)-approximate \(k\)-summation. Actuator location (top, left); feedforward controls from \((N, \xi)\)-approximate \(k\)-summation (bottom, left); desired velocity field and velocity contour for flatness–based state parametrization using (19) (top, middle and right) and respective simulation result (bottom, middle and right).

Fig. 3. Comparison of partial summation (light gray) and \(u_1(t) \mapsto (S_1^{20.1} u_1)(t)\) (black).


