In order to numerically solve the minimization problem (**??**), we follow the general lines of the discretized problem presented in [?gjthesis].

Given $m \in \mathbb{N}_+$, divide the interval $[0,1]$ to m subintervals $I_{m,k} := (\frac{k}{m}, \frac{k+1}{m})$, $k = 0, \ldots, m-1$, of length $\frac{1}{m}$, and consider only H^1-functions that are linear on each subinterval,

$$W_m := \left\{ u \in H^1([0,1], \mathbb{R}^{2n}) \;\middle|\; \dot{u}|_{I_{m,k}} \text{ is constant, } \int_0^1 u(t)dt = 0, \int_0^1 \langle -J\dot{u}, u \rangle = 1 \right\}, \tag{1}$$

where $J \in \mathrm{Mat}(2n, \mathbb{R})$ is the standard complex structure on $\mathbb{R}^{2n}$,

$$J = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}.$$

Let $u \in W_m$, then $\dot{u}$ is piece-wise constant, and can be represented by a vector $\dot{\mathbf{x}} = (\dot{x}_1, \ldots, \dot{x}_m) \in \mathbb{R}^{2n \cdot m}$. In this case, the functional to minimize (**??**) takes the form

$$F(\dot{\mathbf{x}}) = \frac{1}{m} \sum_{j=1}^{m} G(-J\dot{x}_j), \tag{2}$$

and its gradient is

$$\nabla F(\dot{\mathbf{x}}) = \frac{1}{m}(J\nabla G(-J\dot{x}_1), \ldots, J\nabla G(-J\dot{x}_m)). \tag{3}$$

The constraints are given by

$$0 \quad = \quad \ell(\dot{\mathbf{x}}) = \sum_{j=0}^{m-1} \dot{x}_j, \tag{4}$$

$$0 \quad = \quad q(\dot{\mathbf{x}}) = \frac{1}{m^2} \sum_{k=1}^{m-1} \sum_{j=0}^{k-1} \langle -J\dot{x}_k, \dot{x}_j \rangle - 1 = \frac{1}{m^2} \dot{\mathbf{x}}^T A \dot{\mathbf{x}} - 1, \tag{5}$$

where

$$A := \begin{pmatrix} 0_{2n} & -J_{2n} & \cdots & -J_{2n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & -J_{2n} \\ 0_{2n} & \cdots & \cdots & 0_{2n} \end{pmatrix} \in \mathrm{Mat}(\mathbb{R}, 2n \cdot m),$$

and the gradient of the quadratic constraint is

$$\nabla q(\dot{\mathbf{x}}) = \frac{1}{m^2}(A + A^T)\dot{\mathbf{x}}.$$

To minimize the functional F under the constraints ℓ, q we use the Matlab library function named "fmincon". This function recives as input the functional, its gradient, the constraints and their gradients, together with a starting point in $\mathbb{R}^{2n \cdot m}$ and iteratively searches for a local minimum. The function "fmincon" enables to choose a minimization algorithm out of a given list. After numerically experimenting the possible algorithms, we chose "active-set" as it produced the best results in terms of accuracy and run-time. To choose a starting vector, we randomly pick a vector in $\mathbb{R}^{2n \cdot m}$, then shift and rescale it to satisfy conditions (4), (5).