University of BRISTOL

Collocation for control-based continuation

Mark Blyth

Summary

- Previous results: discussing how we could use BSplines for discretisation
  - Lower-dimensional discretisations mean more speed
- Results: it's incredibly numerically finicky
  - Numerical solvers struggle to get accurate results
  - Takes a lot of trial and error to get close to a correct solution curve
- This time: can we use an alternative discretisation?

CBC goal

Say we have a system $\dot{y} = g(y, \mu)$

- Let $y_s(\cdot/T_s)$ be a T_s-periodic solution, for parameter μ_s
- Goal: given (y_0, μ_0, T_0), trace out solution family $\Gamma = \{(y_s, \mu_s, T_s) : s \in \mathbb{R}\}$
- We seek noninvasive control targets to achieve this
  - Noninvasive = target that can be tracked with zero control action
  - Zero control action = system operating under its free dynamics

Finding noninvasive control

- The controlled system maps control target y_{in} to output y_{out}
- Denote this input-output, or IO map, by $Y(y_s, T_s, \mu_s)$
- Assume a proportional controller
- A fixed-point of Y is noninvasive
  - Fixed point means system output exactly matches control target
  - Zero tracking error means zero proportional feedback
  - Zero proportional feedback means controller is switched off

We can use the IO map fixed-point-problem for our continuation equations!

Solving the IO map

- We want to solve the fixed-point problem $y_s^* = Y(y_s^*, T_s, \mu_s)$
  - IO map is evaluated by running the controlled system; slow!
- Continuous problem; not numerically tractable
  - To apply standard numerical methods, we must first discretise the system
- We seek some discretisation of the map
  - Goal: find a finite-dimensional problem that we can pass to a numerical solver
  - Select a discretised problem that also solves the continuous problem

Discretisation with Galerkin projections

Discretisation method used by all current CBC applications

- Take some signal $y(t) \in \mathbb{R}$
- Let β_i, $B_i(t)$, $i = -n, \dots, 0, \dots, n$ be the coefficients, basis functions of its $n-$truncated Fourier series
- β_i is our discretisation, and $\tilde{y} = \sum_{-n}^{n} \beta_i B_i(t)$ is our reconstruction of y
- If $\tilde{y} = Y(\tilde{y}, T_s, \mu_s)$, then $\beta_i^{\text{in}} = \beta_i^{\text{out}}$
- To solve the fixed-point problem, we find the basis function coefficients that remain unchanged when passed through the IO map
  - $2n + 1$-dimensional problem; numerically tractable!

Issues with current CBC discretisation

- Evaluation of continuation equations is *slow*
  - Newton iterations require a Jacobian, which requires finite differences
  - This means we need to run physical system to convergence, many times

- We can only find the noninvasive solution y using Galerkin discretisation when $y \in span\{B_1, B_2, \ldots B_m\}$
  - This limits our choice of basis functions
  - We might still be able to find an approximate solution when this doesn't hold, but I wouldn't know how to prove or disprove this

BSpline discretisation

- We can speed up prediction-correction by reducing number of evaluations
  - Easily achievable with lower-dimensional discretisation
- One option: use more ‘efficient’ basis functions
  - A Fourier basis is inefficient for neuronal signals; can we find more efficient basis functions?
- Discretisation with BSplines is very numerically difficult; hard to find an accurate solution, even when playing with
  - Continuation stepsize
  - Finite differences stepsize
  - Number of basis functions
  - Convergence tolerance
- Another option: can we use non-Fourier basis functions with another discretisation method?

Collocation

Instead of solving the problem exactly, by requiring the input and output discretisations to be exactly equal, we could solve it approximately

- Collocation defines a discrete approximation of the problem, that we can solve exactly
- We can always find an approximate solution when using collocation *[I think]*
  - Collocation solution will *[hopefully]* be easier to find
  - Conjecture: the collocation solution will be identical to the Galerkin projection solution in cases where Galerkin projection works
- Collocation discretisation – hopefully less numerically fiddly
- Non-fourier basis functions – lower-dimensional discretisation: faster!

Collocation setup

- We approximate the solution with $y_{\text{in}} = \sum \beta_i B_i(t)$, for some basis functions $B_i(t)$
- We split the signal period into a mesh $[\xi_1 = 0 < \xi_2 < \cdots < \xi_n = T_s]$
- We solve for β_i such that $y_{\text{in}}(\xi_i) = y_{\text{out}}(\xi_i)$
  - We also add any phase constraints, periodicity constraints into the system
  - Here, $y_{\text{out}}(t) = Y(\sum \beta_i B_i(t), T_s, \mu_s)$
- Collocation solution solves the fixed-point problem exactly, at the collocation meshpoints
  - We assume it's a good approximation between meshpoints
  - Resulting β_i give our signal discretisation for continuation
  - Resulting function $\sum \beta_i B_i(t)$ gives a control target

Comparison of methods

- Galerkin methods require us to translate from signal to discretisation, eg. using FFT
- Collocation does not, which offers a slight speed-up
- Galerkin basis functions will help filter noise off
- Collocation offers no noise-filtering
- Collocation automatically aims to find the best approximation; it should hopefully be robust to cases where a solution can only be found to a limited accuracy
- Galerkin methods aim for a correct solution straight away; this makes them harder to apply when we're limited in solution accuracy, eg. by not having enough Fourier harmonics

Potential collocation pitfalls

- Collocation is not noise-robust
  - We're searching for equality between input and output signals
  - If we have measurement error, output values at the meshpoints are a random variable
  - Instead of searching for equality, we would need a maximum-likelihood estimation on β_i
  - Alternatively, use a surrogate model to filter the noise off!
- Collocation finds an approximate solution
  - We can only guarantee the discretised problem to be solved at the meshpoints
  - Collocation solution may deviate from true solution between meshpoints, in which case we wouldn't have noninvasive control
  - Could implement a solution-checker, by measuring the distance between input and output functions

Next steps

- Code up a collocation CBC simulation
- See if it works!
- Perform numerical experiments to compare its solution accuracy against Galerkin discretisation
- Test its noise-robustness with surrogates