Discretisation-free CBC

Mark Blyth

Week’s goal

- Finish off redrafting paper
- Start working towards an *in-silico* CBC

Week's activities

- Finished off redrafting paper
- Started reading a paper for a single-cell model to test CBC on [1]
  - Krasi's cubic Lienard model, but with a parameter fixed, and coupled to a slow subsystem
  - Capable of modelling almost all known bursting behaviours
- Read some of Kuznetsov numerical bifurcation analysis
- Started thinking about CBC
  - This week's big idea: discretisation-free CBC

[1] Saggio, Maria Luisa, et al. "Fast–slow bursters in the unfolding of a high codimension singularity and the ultra-slow transitions of classes." The Journal of Mathematical Neuroscience 7.1 (2017): 7.

Continuation background: points

- Continuation works in a predictor corrector scheme
  - Predict the next point on the manifold from the local tangent vector
  - Correct it using a Newton iteration
  - An additional parameter appears – the arclength parameter – so require *predictor* $\perp$ *corrector* to ensure a well-posed problem
- For equilibrium and equilibrium-bifurcation continuation, we have a finite-dimensional state
  - Tangent vector is of the same dimensionality, and is therefore finite
  - Predictor-corrector scheme is of finite – usually low – dimensionality, and is therefore computationally tractable

For points (equilibria, equilibrium bifurcations) everything works nicely

Continuation background: orbits

- A periodic orbit is some function $f(t, \lambda),\ t \in [0, 1]$
  - f exists in an infinite-dimensional Hilbert space
- Continuation of f in λ requires a discretisation, to produce a finite-dimensional approximation that we can apply standard continuation methods on
- There's a range of methods for discretisation
  - Orthogonal collocation seeks a set of orthogonal polynomials that satisfy the model at a selection of meshpoints; high accuracy; requires a model
  - Fourier decomposition decomposes a periodic signal into its harmonic components; model-free (important for CBC); sensitive to noise; will be high-dimensional for spiking signals
  - Wavelets, frames, splines, . . . , yet to be developed!

Issues with discretisation

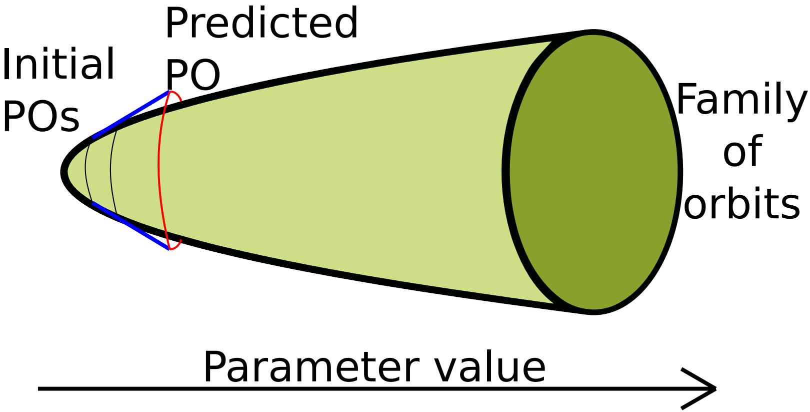
- Can't use collocation methods without a model
- Spiking signals would need a lot of Fourier harmonics (quick-changing means lots of high-frequency energy); high dimensional continuation systems are hard
- Noise would greatly impair Fourier discretisation; can't filter it off without losing the high-frequency components of the signal required for fast spiking
- Wavelets, frames, splines haven't been developed yet (might also be noise-sensitive?)

Can we continue periodic orbits without discretisation?

Discretisation-free method: benefits and issues

- By avoiding discretisation, we can deal with fast-changing signals easily
- The learning step allows us to average out the noise, in a way that would be difficult using discretisation methods, meaning more numerical stability
- Uses some machine learning – a buzzword that seems to bring in citations…
- Fourier is a more natural discretisation choice for periodically excited systems
    - If we can partition the control action into a controller and a periodic forcing term, it makes sense to do so
    - For neurons, where the stimulus and output are different, we can't do this partitioning, so we lose the benefits of Fourier

Graphic representation



Basic strategy

- Learn a local model of the periodic orbit surface
- Use that model to predict the next periodic orbit
  - Learning and projecting forms the predictor step
- Take the learned, predicted PO model as the control target
- Iteratively update it, orthogonally to the forward-projection
  - This iterating forms the corrector step

No need to discretise the signal, as we fit a continuous model to the data and work from that instead.

- The zero problem is given by the noninvasive control requirement, rather than from a model
- This means that we don't actually need to find a discretisation of the periodic orbit, unlike in model-based zero-problems

A topology interlude

- A homotopy H is a continuous deformation $H : X \times [0, 1] \to Y$ between two topological spaces X and Y
- Consider a homotopy H between functions f_1, f_2, parameterised in some variable t
  - $H(f_1, 0) = f_1$
  - $H(f_1, 1) = f_2$
  - Simple example: $H = f_1 + t(f_2 - f_1)$
- Animation 1
- Animation 2

The overall goal is to learn a continuous homotopic transformation for the predictor/corrector, which can be applied to raw, undiscretised data

Mathematical representation

- Use machine learning to find a homotopy between successive orbits $f(t, \lambda_{i-1})$, $f(t, \lambda_i)$
- Use this homotopy as a predictor for the next orbit
- Apply an orthogonal correction step
  - Prediction will be a smooth function estimating $f(t, \lambda_1)$
  - Find a corrector family of f orthogonal to the homotopic step
  - Each f in this family is a control target, one of which is a periodic orbit of the open-loop system
  - ‘Slide down’ this family of periodic orbits, on to the corrected solution
  - ‘Sliding down’ is done by iteratively updating the control target, much like in Barton et al.
  - By selecting new targets from the corrector family, we’re maintaining the orthogonality constraint

Learning a homotopy

1. Set $\lambda = \lambda_0$
2. Record data for a while
3. Use F_0 estimator to partition data into periods
4. Reconstruct the state space (?)
5. Let $t \in [0, 1]$ measure how far through a period each reconstructed vector is
6. Learn a function $f_0 : [0, 1] \to \mathbb{R}^n$, giving the (reconstructed) state at time t
7. Repeat this for $\lambda = \lambda_1$, learning function f_1
8. Learn a homotopy $H_1 : \mathcal{H} \times [0, 1] \to \mathcal{H}$, where $f_i \in \mathcal{H}$

The machine learning step

- Gaussian processes are the ideal tool for learning f_i, H_i
  - Provide a nonparametric way of modelling arbitrary manifolds
  - Statistically rigorous
- F_0 estimation and state space reconstruction is much like that in my master's thesis
- Might even be able to get away without the state space reconstruction, but intuitively it seems like everything would work better doing it

Benefits and issues (again)

- By avoiding discretisation, we can deal with exteedingly fast-changing signals easily
- The learning step allows us to average out the noise, in a way that would be difficult using discretisation methods, meaning more numerical stability
- Fourier is a more natural discretisation choice for periodically excited systems
  - If we can partition the control action into a controller and a periodic forcing term, it makes sense to do so
  - For neurons, where the stimulus and output are different, we can't do this partitioning, so we lose the benefits of Fourier
- Prediction step should be fairly straightforward
- Correction step *might* be straightforward, but has the potential to be more challenging

Next steps

- Finish readings (Kuznetsov numerical bifurcation analysis, neuron model paper)
- Make any additional changes to the continuation paper
- Further programming marking
- Lab meeting Wednesday; make some slides for that
  - Current plan: present everything I've written in the paper
  - Nb. I have managed to get Zoom to work, but can't use Skype for business
- Try implementing Fourier CBC for a neuron
- Adapt that for discretisation-free CBC