Goals for this lecture

- Numerical Analysis
  - Finding Fixed Points
  - Local Stability
  - Basins of Attraction
- Robots based on simple models

1 Introduction

We’ve looked at a number of simple models of legged locomotion, and spent time with the ones that are analytically tractable. Today I’d like to wrap up those discussions by looking at numerical tools for models that are not analytically tractable, and by looking at the robots that we designed using these models.

2 Numerical Analysis

A note on integrating hybrid systems and/or evaluating return maps. The dynamics are often very sensitive to the switching plane. Often a good idea to back up integration to attempt to find collisions and transitions very accurately.

2.1 Finding Fixed Points

You can find asymptotically stable fixed by integrating forward the ODE (until $t \to \infty$). This convergence rate will depend on the convergence rate around the fixed point and could be inefficient for complex systems. This method won’t work for finding unstable fixed points.

**Newton-Raphson Method.** Remember that a fixed point of the nonlinear system

$$\dot{x} = f(x),$$
is simply a zero-crossing of the (potentially vector-valued) function $f$. Therefore, another way to obtain the fixed points numerically is to use an algorithm which finds the roots of $f$. If $x$ is one-dimensional, then many algorithms (including `fzero' in MATLAB) can efficiently find these zero-crossings. If $x$ is multi-dimensional, then the Newton-Raphson method will work for you, assuming you have a decent initial guess\[4\]. In general, finding multi-dimensional zero crossings is extremely difficult. Newton-Raphson uses the gradient of $F$ (given analytically, or estimated numerically) to make an update

$$x_{i+1} = x_i - \left[ \frac{\partial f}{\partial x} \right]^{-1}_{x_i} f(x_i).$$

Graphically, this takes the point where the tangent line of the function at $x_i$ crosses the origin. Draw cartoon on the board. Estimating the gradient numerically can be a pain, but this method will work equally well for stable and unstable fixed points. And it works equally well for discrete-time systems (e.g. on a Poincaré map).

### 2.2 Local Stability of Limit Cycle

In practice, the local stability analysis of a limit cycle is done by taking the derivatives around the fixed point of the return map. Again, this is often accomplished using numerical derivatives. Perturb the system in one direction at a time, evaluate the map and build the matrix ... From Goswami \[3\]. The eigenvalues of the derivative matrix of the Poincaré map, $\lambda_i$ are called the characteristic or Floquet multipliers\[7\].

$$x^* + \delta_1 = P(x^* + \delta_0) \approx P(x^*) + \left[ \frac{\partial P}{\partial x} \right]_{x^*} \delta_0.$$  

$$\delta_1 \approx \left[ \frac{\partial P}{\partial x} \right]_{x^*} \delta_0.$$  

A fixed point is stable if the $n-1$ non-trivial eigenvalues of this matrix are $|\lambda_i| < 1$.  

Trivial multipliers vs. Non-trivial multipliers. Expect one trivial multiplier of 0, or 1 (which reveal the dynamics of a perturbation along the limit cycle orbit).

A standard numerical recipe for estimating $\frac{\partial P}{\partial x}$ is to perturb the system by a very small amount at least $n$ times, once in each of the state variables, and watching the response. Be careful - your perturbation should be big enough to not get into integration errors, but small enough that it stays in the "linear regime". A good way to verify your results is to perturb the system in other directions, and other magnitudes, in an attempt to recover the same eigenvalues. In general, the matrix $\frac{\partial P}{\partial x}$ can be reconstructed from any number of sampled trajectories by solving the equation

$$\begin{bmatrix} \delta^1_0 & \delta^2_0 & \cdots & \delta^m_0 \\ \delta^1_1 & \delta^2_1 & \cdots & \delta^m_1 \\ \vdots & \vdots & \ddots & \vdots \\ \delta^1_m & \delta^2_m & \cdots & \delta^m_m \end{bmatrix} = \left[ \frac{\partial P}{\partial x} \right]_{x^*} \begin{bmatrix} \delta^1_0 & \delta^2_0 & \cdots & \delta^m_0 \\ \delta^1_1 & \delta^2_1 & \cdots & \delta^m_1 \\ \vdots & \vdots & \ddots & \vdots \\ \delta^1_m & \delta^2_m & \cdots & \delta^m_m \end{bmatrix}$$

in a least-squares sense, where $\delta^i_0$ is the $i$-th perturbation (not a perturbation raised to a power!).
Lyapunov exponent. There is at least one quantifier of limit cycle (or trajectories, in general) stability that does not depend on the return map. Like a contraction mapping - perturb original trajectory in each direction, bound recovery by some exponential\[7\].

\[
\|\delta(t)\| < \|\delta(0)e^{At}\|.
\]

The eigenvalues of \(A\) are the Lyapunov exponents. Note that for a stable limit cycle, the largest Lyapunov exponent will be 1 (like the trivial floquet multiplier), and it is the remaining exponents that we will use to evaluate stability.

2.3 Basins of Attraction

For your problem set, you computed the basin of attraction for one of the fixed points on the simple pendulum. You probably ran, from each initial condition, until it entered some small neighborhood of one of the stable fixed points. You also probably thought to yourself, while the computer was chugging away, that there has to be a better way.

Cell Mapping Method Finely discretize space, and start by labelling cells that are clearly out of the basin of attraction (sink cells - can be few and large). The start making transitions between cells. Stop as soon as you enter a known cell, or encounter a loop. All trajectories end in either a known cell or the sink cells[6]. This method is only as accurate as your discretization. But it allows you to find fixed points, simple limit cycles, and multi-step limit cycles.

3 Robots based on simple models

3.1 Passive Dynamic Walkers since McGeer

Collins walker (again). Wisse walkers.

3.2 Actuated Walkers based on PDWs

Describe three robots[2] and show the videos.

Possibly shelve this until another lecture: Evaluating dynamics of the physical machine. True fixed-point doesn’t exist (world is always changing, it’s a non-stationary, stochastic process). Hard to estimate true outline of basin of attraction (although we can sample from it). Can still try to find local stability. Find approximate Least-squares exponential fit. From my PDW paper[8].

3.3 More Raibert Machines

Uniroo. Quadruped hopper (running on four legs as though they were one)[5]. Big Dog.

3.4 RHex

Describe RHex[1] and show videos.
References


