Dynamic Movement Primities (Ijspeert, 2013)

(Willow Garage video)

Course so far

1) Write down equations of motion + doing computation simulation

2) Model learning

3) Planning: RRTs

4) Complex movements

- general way to represent movements (DMPs)
- way to learn movements (learning from demonstration)
- way to automatically improve (reinforcement learning)

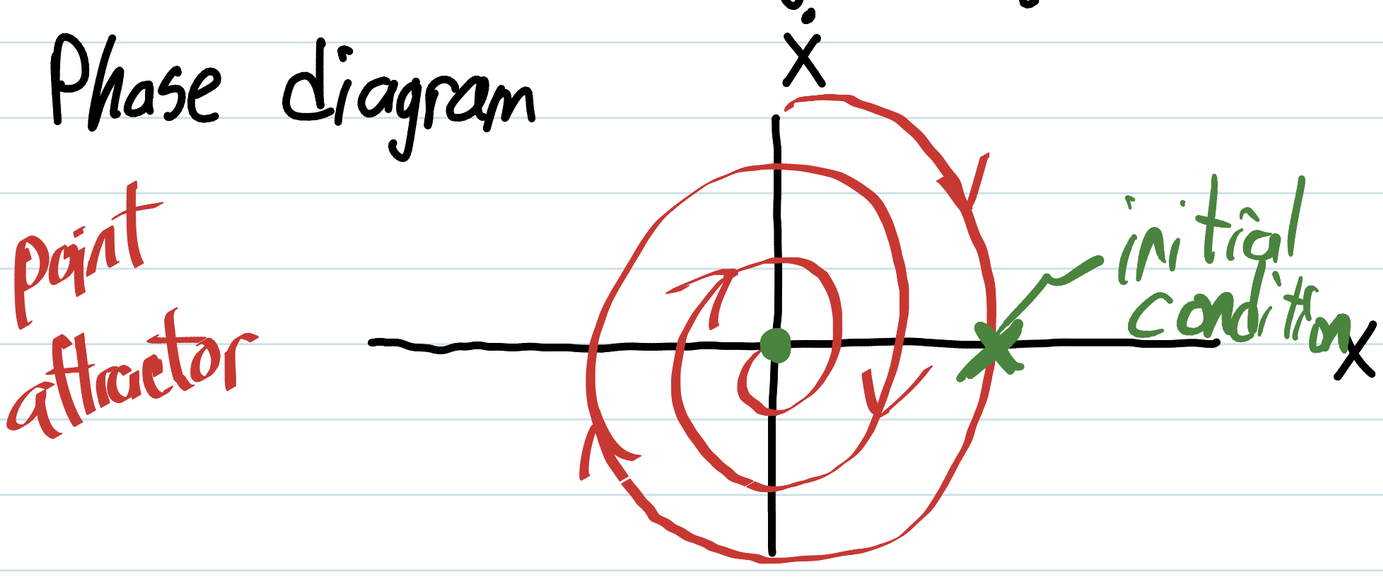
DMPs are a general description of complex movements that does these things:

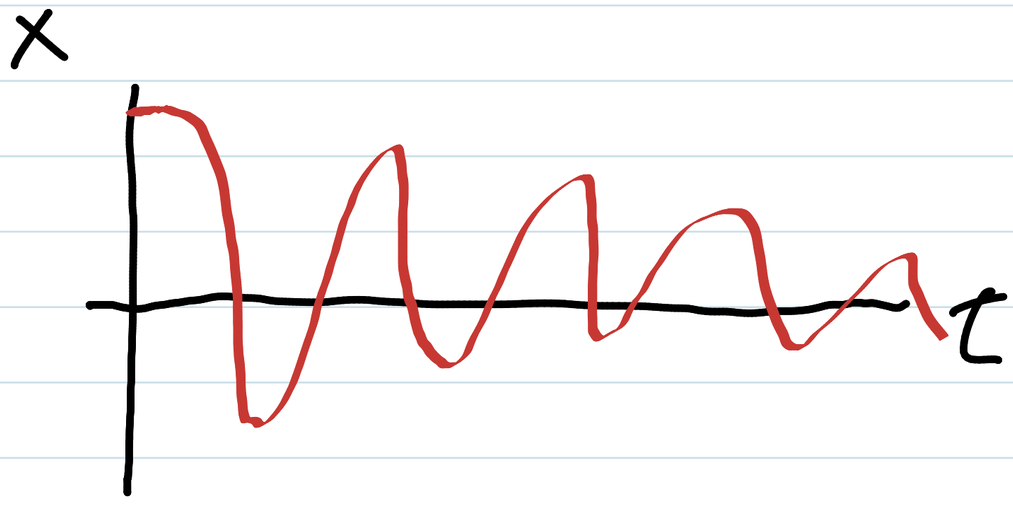
1) represent learnable point attractor and limit cycle attractors. point attractor defines pt-to-pt movements
limit cycle defines cyclic movements

e.g. $m\ddot{x} + c\dot{x} + kx = 0$

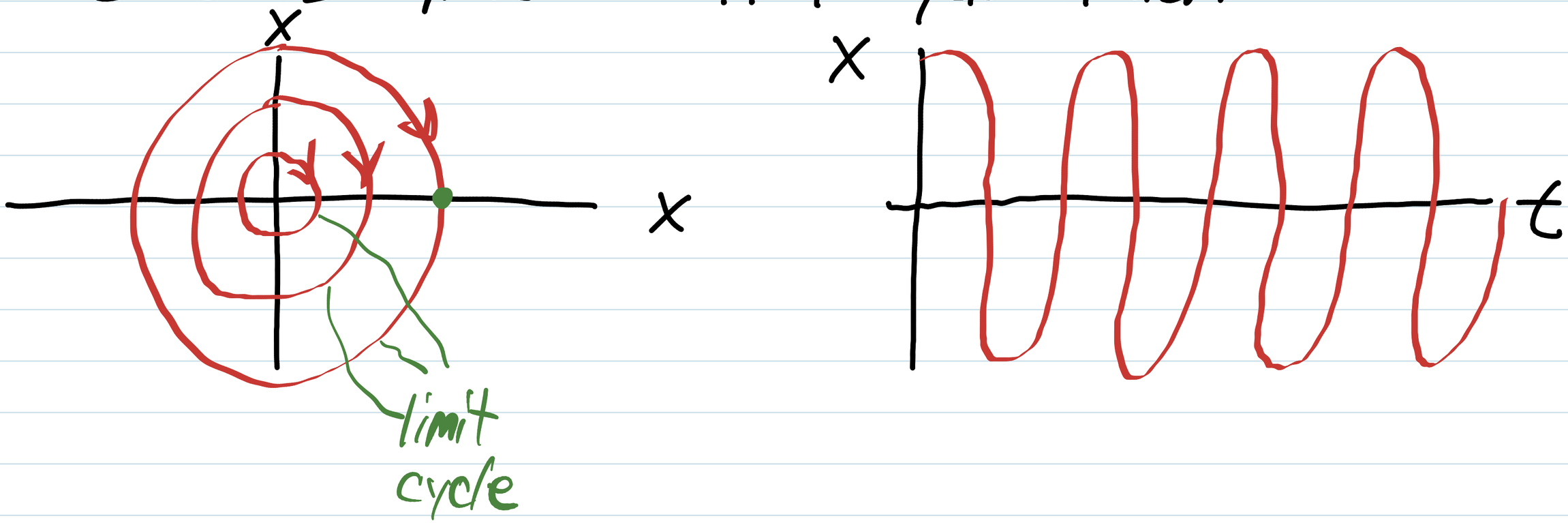
mass-spring-damper

Phase diagram





if $c = 0$ I have a limit cycle attractor



2) The model should be an autonomous system with no explicit time dependence. ie. the phase diagram looks the same no matter what the time is.

3) The model needs to be able to coordinate multidimensional dynamical systems in a stable way.

4) Learning model parameters should be simple (e.g. linear in parameters)

5) Model needs to incorporate coupling terms

6) Model should allow real-time computation and modulation of trajectories (real-time planning)

7) Model should have spatial scale and time invariance. Changing how far to travel or how fast shouldn't change the geometry of the attractor landscape.

Here's the DMP model: 2nd order spring-mass-damper system with a nonlinear forcing term

$$\tau \ddot{y} = \alpha_z (\beta_z (g - y) - \dot{y}) + f$$

y is the position of the system (robot hand position)

g is the goal position (position of the cup)

τ is a time constant (like inertia)

α_z is a constant related to damping

β_z is a constant related to stiffness

f is a nonlinear forcing term

We can write the DMP model as two first-order systems

$$\tau \dot{z} = \alpha_z(\beta_z(g-y) - z) + f$$
$$\tau \dot{y} = z$$

] transformation system

If $f = 0$ this is a stable second-order linear system with a unique point attractor at $(z, y) = (0, g)$

you stop at the goal

You can pick α_z and β_z so that the system is critically damped e.g. ($\beta_z = \alpha_z/4$). This makes the system move monotonically toward g without oscillation.

This is a stable but trivial point attractor - just one (for $f=0$)

geometry that can be modulated with τ, α_z, β_z.

If $f \neq 0$ you can define an arbitrary attractor geometry.

What does this DMP model do?

It tells you $[y, \dot{y}, \ddot{y}]$ which is a trajectory you want to follow by using computed torque + feedback controller.

Adjusting the attractor landscape with the nonlinear forcing function.

forcing function $f(t) = \dfrac{\sum_{i=1}^{N} \Psi_i(t) w_i}{\sum_{i=1}^{N} \Psi_i(t)}$

N fixed basis functions $\Psi_i(t)$

w_i adjustable weights

learn the weights via learning from demonstration

$f(t)$ changes the attractor landscape as a function of time.

Let's replace time with a phase variable x which obeys 1st-order dynamics

$$\tau \dot{x} = -\alpha_x x$$ canonical system

This system has a solution

$$x(t) = x_0 e^{-\frac{\alpha_x}{\tau} t} \qquad x_0 = x(t=0) = 1$$

Decaying exponential that converges to $x=0$

$x=1$ represents the start of the trajectory

$x=0$ represents the end

Restate nonlinear forcing term as a function of the phase variable x instead of time t

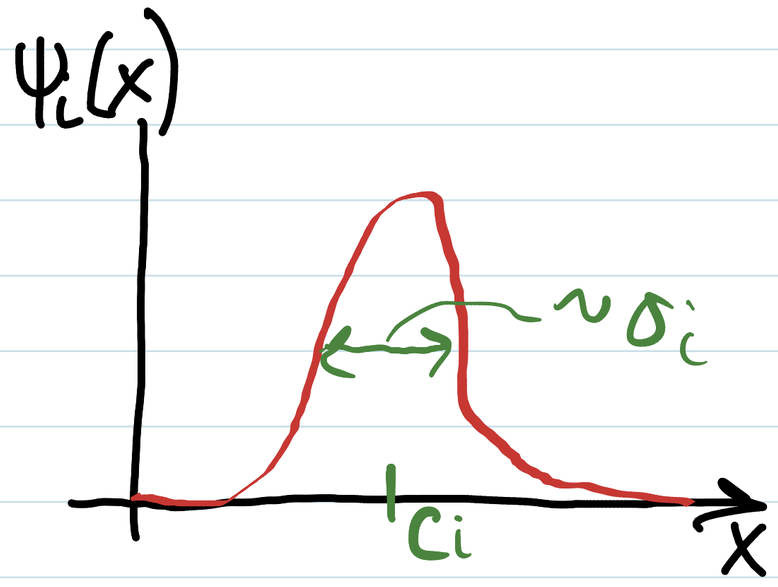
$$f(x) = \frac{\sum_{i=1}^{N} \psi_i(x) w_i}{\sum_{i=1}^{N} \psi_i(x)} x(g - y_0)$$

N squared exponential basis functions

$$\psi_i(x) = \exp\left(-\frac{1}{2\sigma_i^2}(x - c_i)^2\right)$$

c_i is center of basis fcn

σ_i is the width



X is how far you are along a trajectory, so the basis functions are active at different points in the trajectory.

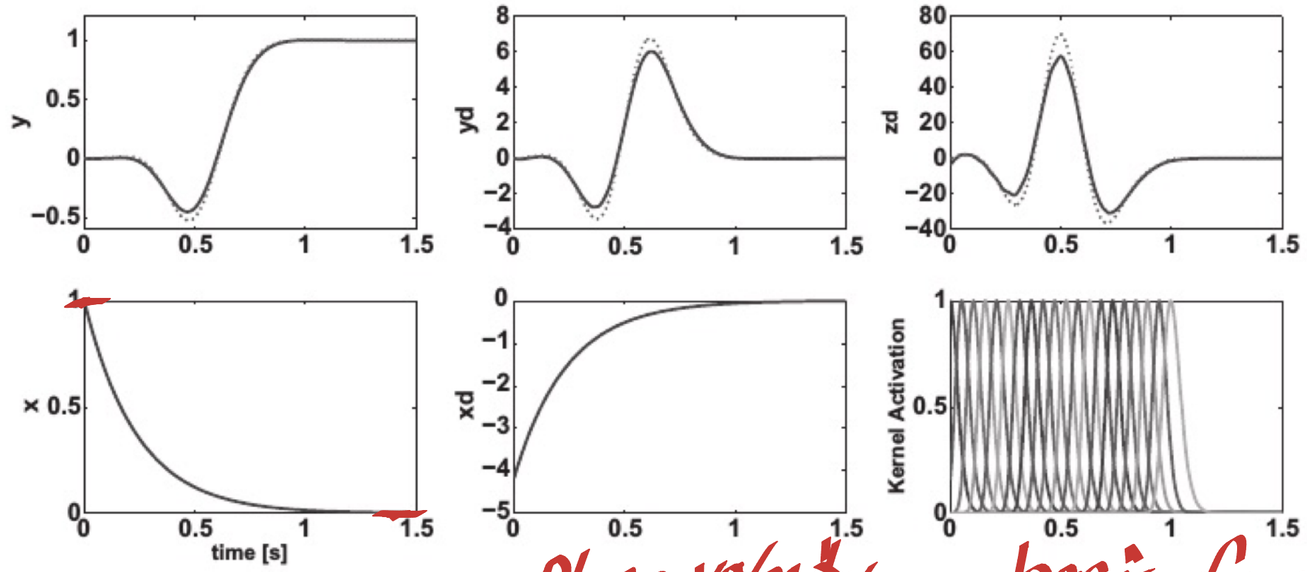
The forcing term $f(x)$ is modulated by

- X which means the forcing term vanishes ($x=0$) When the goal is reached
- $g - y_0$ Which gives nice spatial scaling properties (forcing function is different when total distance between start and goal is different)

The entire system has a stable fixed point at

$$(z, y, x) = (0, g, 0)$$

velocity ← "position" phase



phase

phase velocity

basis functions $\psi_i(x)$

Figure 1: Exemplary time evolution of the discrete dynamical system. The parameters w_i have been adjusted to fit a fifth-order polynomial trajectory between start and goal point ($g = 1.0$), superimposed with a negative exponential bump. The upper plots show the desired position, velocity, and acceleration of this target trajectory with dotted lines, which largely coincide with the realized trajectories of the equations (solid lines). On the bottom right, the activation of the 20 exponential kernels comprising the forcing term is drawn as a function of time. The kernels have equal spacing in time, which corresponds to an exponential spacing in x.

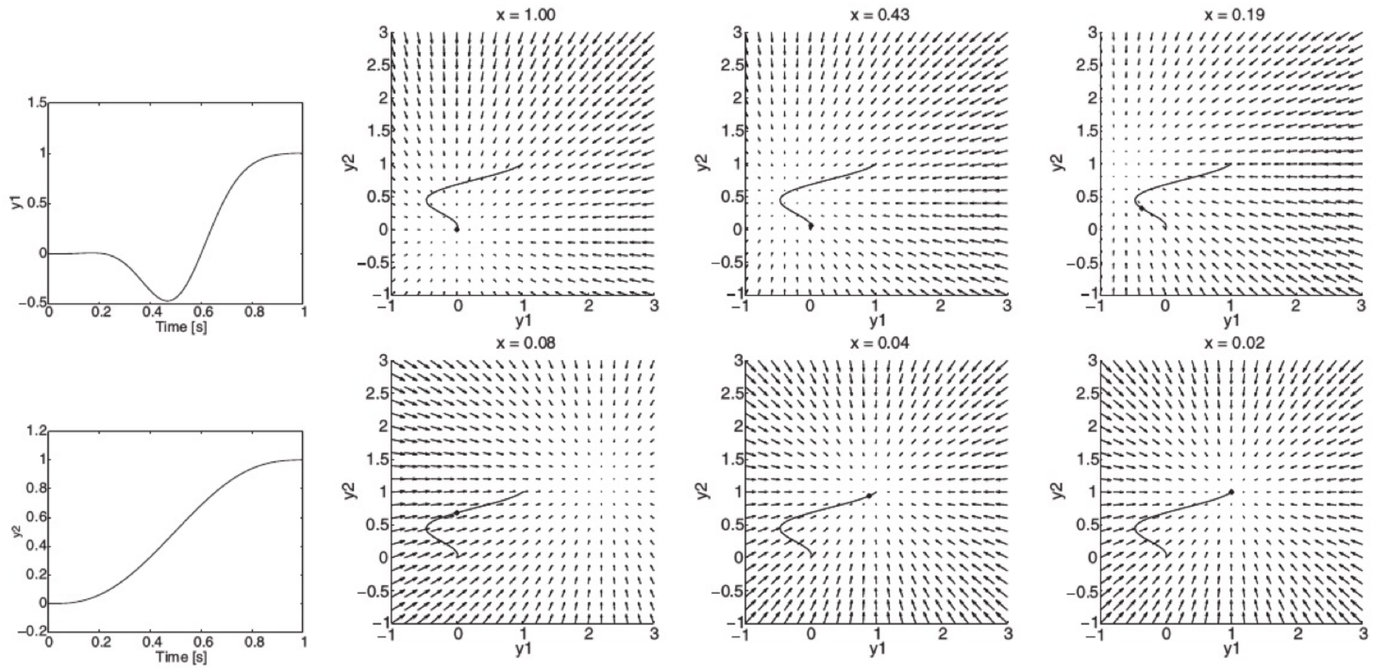


Figure 2: Vector plot for a 2D trajectory where y_1 (top left) fits the trajectory of Figure 1 and y_2 (bottom left) fits a minimum jerk trajectory, both toward a goal $g = (g_1, g_2) = (1, 1)$. The vector plots show $(\dot{z}_1, \dot{z}_2)$ at different values of (y_1, y_2), assuming that only y_1 and y_2 have changed compared to the unperturbed trajectory (continuous line) and that x_1, x_2, $\dot{y}_1$, and $\dot{y}_2$ are not perturbed. In other words, it shows only slices of the full vector plot $(\dot{z}_1, \dot{z}_2, \dot{y}_1, \dot{y}_2, \dot{x}_1, \dot{x}_2)$ for clarity. The vector plots are shown for successive values of $x = x_1 = x_2$ from 1.0 to 0.02 (i.e., from successive steps in time). Since $\tau \dot{y}_i = z_i$, such a graph illustrates the instantaneous accelerations $(\ddot{y}_1, \ddot{y}_2)$ of the 2D trajectory if the states (y_1, y_2) were pushed somewhere else in state space. Note how the system evolves to a spring-damper model with all arrows pointing to the goal $g = (1, 1)$ when x converges to 0.

Invariance properties: DMPs are temporally and spatially invariant to changes in τ (time inertia) and $g - y_0$ (spatial scale)

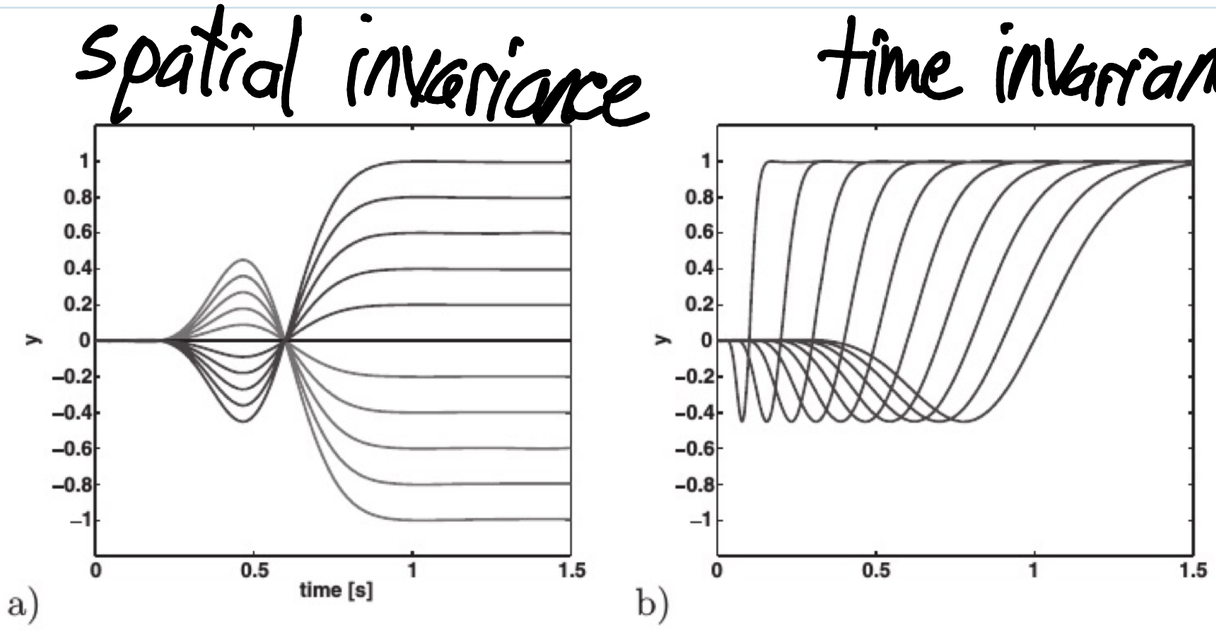


Figure 4: Illustration of invariance properties in the discrete dynamical systems, using the example from Figure 1. (a) The goal position is varied from -1 to 1 in 10 steps. (b) The time constant τ is changed to generate trajectories from about 0.15 seconds to 1.7 seconds duration.

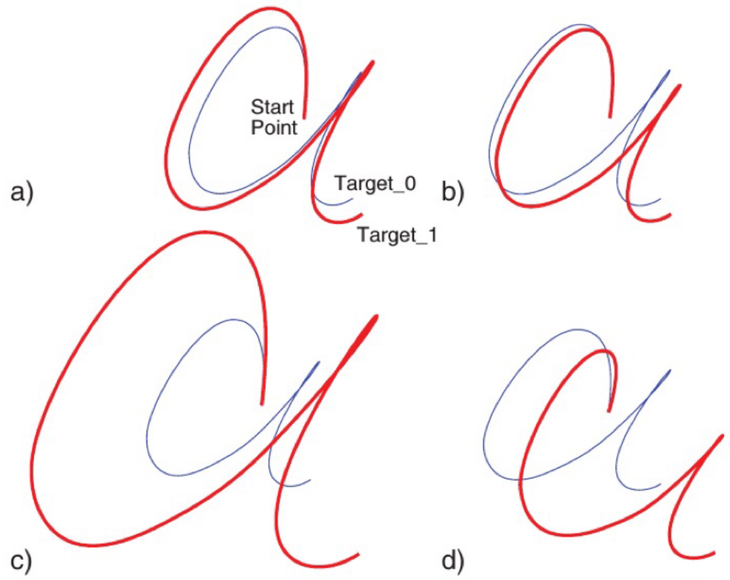


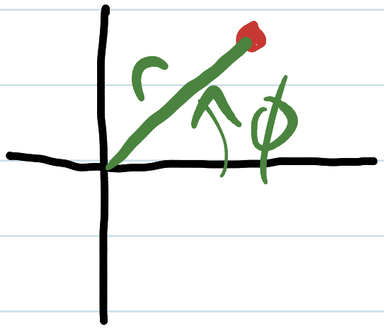
Figure 5: Illustration of the significance of the invariance properties, exemplified in a two-dimensional discrete dynamical system to draw a cursive letter a. In all subfigures, the blue (thin) line denotes the letter "a" as taught from a human demonstration using a digitizing tablet. The start point for all figures is the same, while the goal is originally $Target_0$, and, for the purpose of testing generalization, the goal is shifted to $Target_1$. In *a* and *b*, the shift of the goal is small, while in *c* and *d*, the shift of the goal is much more significant. Subfigures *a* and *c* use equations 2.1 to 2.4, the proper formulation of the discrete dynamical system with invariance properties. As can be noted from the red (thick) lines, the generalized letter "a" is always a properly uniformly zoomed version of the original letter "a." In contrast, in subfigures *b* and *d*, the scaling term $g - y_0$ in equation 2.3 was left out, which destroys the invariance properties. While for a small shift of the goal in *b* the distortion of the letter "a" is insignificant, for a large shift of the goal in d, the distortion creates more a letter "u" than a letter "a."

Adjusting the attractor landscape for limit cycle attractors

Instead of choosing a decaying canonical system, we pick a phase oscillator

$$\tau \dot{\phi} = 1 \qquad \text{where } \phi \text{ the phase angle is } \in [0, 2\pi]$$

and amplitude r



modulating r changes how big the oscillations are

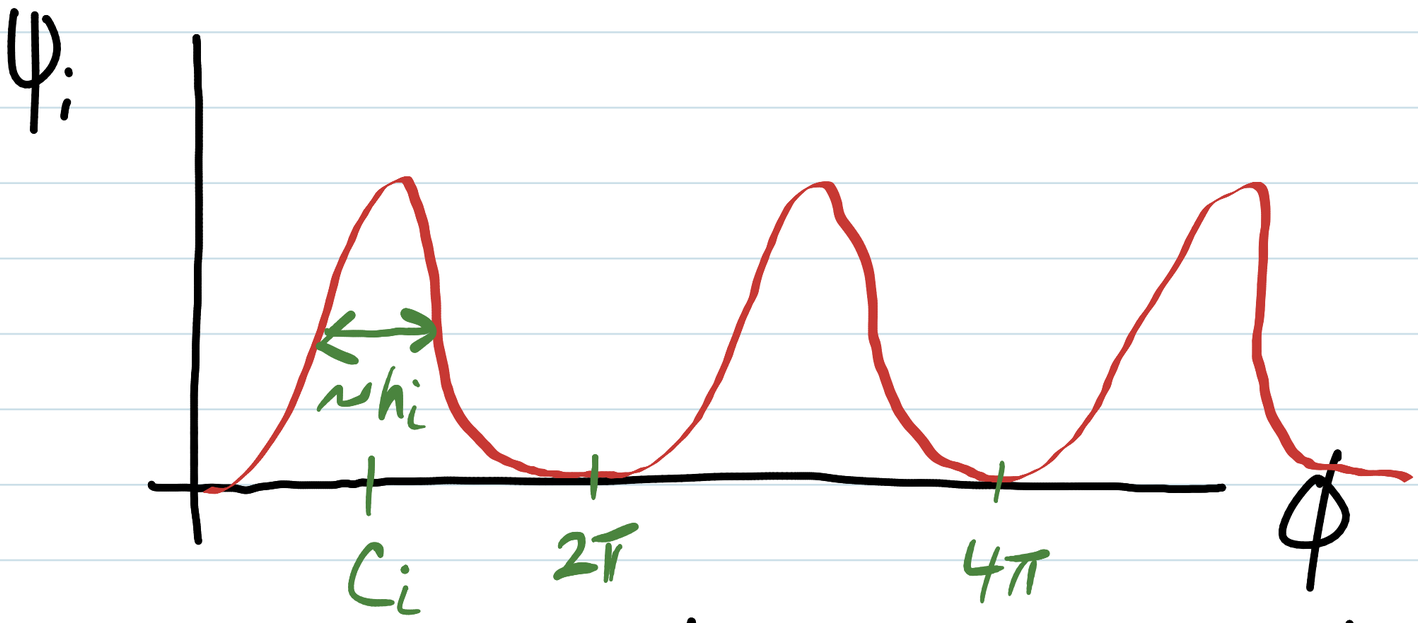
modulating τ changes the period of oscillation

The forcing function is

$$f(\phi, r) = \frac{\sum_{i=1}^{N} \psi_i(\phi) w_i}{\sum_{i=1}^{N} \psi_i(\phi)} r$$

where the basis functions are von Mises basis functions

$$\psi_i(\phi) = \exp\left(h_i\left(\cos(\phi - c_i) - 1\right)\right)$$ (periodic Gaussian)



... find h_i and c_i by regression using demonstration data

If we look back at the transformation system

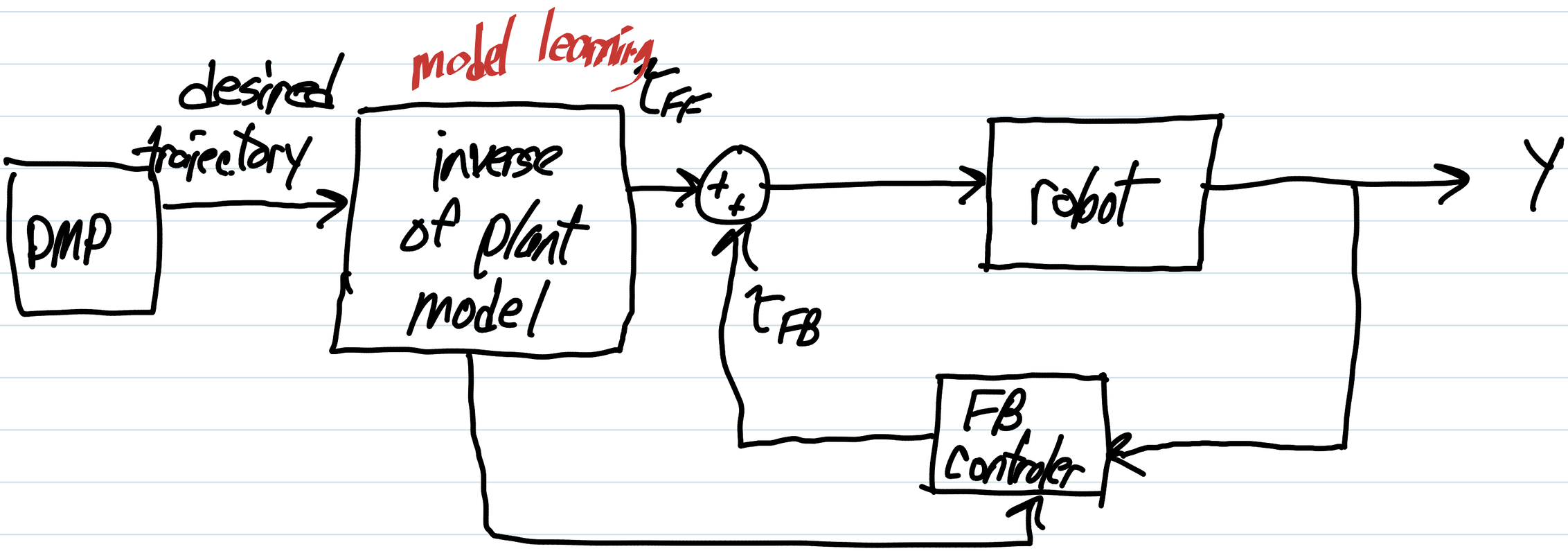
$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f$$

$$\tau \dot{y} = z$$

(Ijspeert Fig 3)

g acts as the anchor point or baseline for the oscillation
The amplitude and period are modulated by r and τ

How do DMPs fit in with other stuff we have done?



Multiple degrees of Freedom

Use one canonical system for all degrees of freedom

$$\tau \dot{x} = -\alpha_x x$$

Use a different transformation system for each DoF

$$\tau \dot{z}_i = \alpha_{z_i}\left(\beta_{z_i}(g_i - y_i) - z_i\right) + f_i(x)$$

$$\tau \dot{y}_i = z_i \qquad \text{for the } i\text{th degree of freedom}$$

Learning attractor dynamics from observed behavior

Goal: Find g, y_0, τ, and w_i from observed movements

observation $t = [0 \dots P]$ $\quad y_d(t)$, $\dot{y}_d(t)$, $\ddot{y}_d(t)$

g is the final position $y_d(t=P)$

y_0 is the initial position $y_d(t=0)$

τ is adjusted to the duration of the demonstration

w_i are found using your favorite regression technique

Lets look at the transformation system

$$\tau \dot{z} = \alpha_z(\beta_z(g-y) - z) + f \qquad \tau \dot{y} = z$$

rearrange

$$f = \tau \dot{z} - \alpha_z(\beta_z(g-y) - z)$$

$$= \tau^2 \ddot{y} - \alpha_z(\beta_z(g-y) - \tau \dot{y})$$

From the training data

$$f_t = \tau^2 \ddot{y}_d - \alpha_z(\beta_z(g - y_d) - \tau \dot{y}_d)$$

f_t: target value, what forcing function should be

$\ddot{y}_d, y_d, \dot{y}_d$: demonstration data

Find parameters by minimizing the difference between f_t and what my model forcing function f predicts

$$\min_{w_i} \sum_{t=0}^{P} \left(f_t - \frac{\sum_{i=1}^{N} \psi_i(x(t)) w_i}{\sum_{i=1}^{N} \psi_i(x(t))} x(t)(g - y_0) \right)^2$$

Online modulation of attractor dynamics