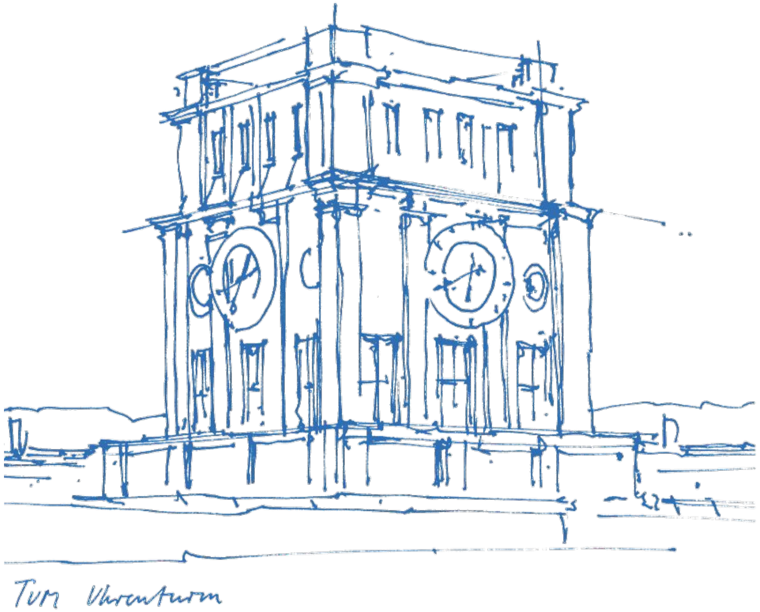
NISE Snake Robot Challenge

Kunal Aggarwal, Katja Frey, Alexandra Samoylova,
Oscar Soto Rivera, Maria Zeller

Elite Master Program Neuroengineering
Technical University of Munich

08.02.2024



Building the Network and Working Out Coordinated Locomotion:

- **Matsuoka oscillator model**: several parameters influence CPG output

  ⇨ robot’s control becomes complicated and difficult

- **Phase Oscillator Model** (Nor & Ma, 2013, Nor & Ma, 2014)

  The paper describes: ✅ turning behavior
  ✅ change in # of S-shapes (motion pattern change)

  ⇨ a promising approach!

Phase Oscillator Model

Mathematical model of Phase Oscillator:

$$\boxed{\tau}\boxed{\dot{\theta}_i} = 2\pi\boxed{\nu_i} + \sum \boxed{w_{ij}}\sin(\boxed{\theta_j} - \boxed{\theta_i} - \boxed{\phi_{ij}})$$

Output of each oscillator:

$$x_i = \boxed{A}\cos(\theta_i)$$

$$goal\_position = x_i + \boxed{offset}$$

→ *solved with Euler Method*

Time evolution of the phase $\boldsymbol{\theta}_i$

Intrinsic frequency

Coupling oscillator *j* to oscillator *i*

Phase of the *j*th oscillator

Phase of the *i*th oscillator

Phase bias

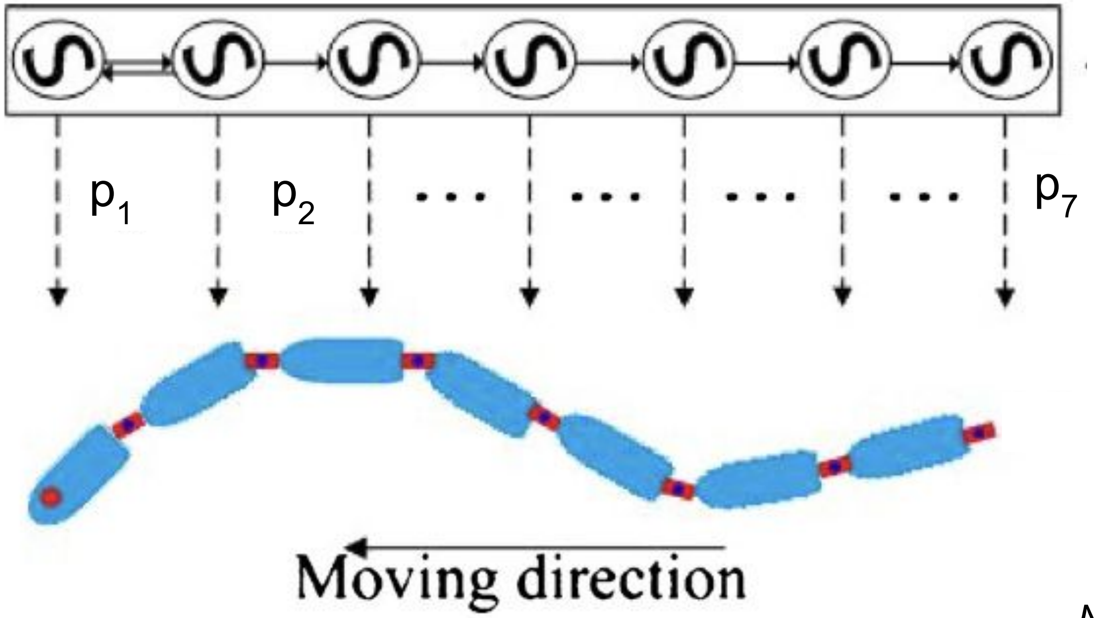
Time constant

Amplitude

Trajectory offset

Phase Oscillator Model

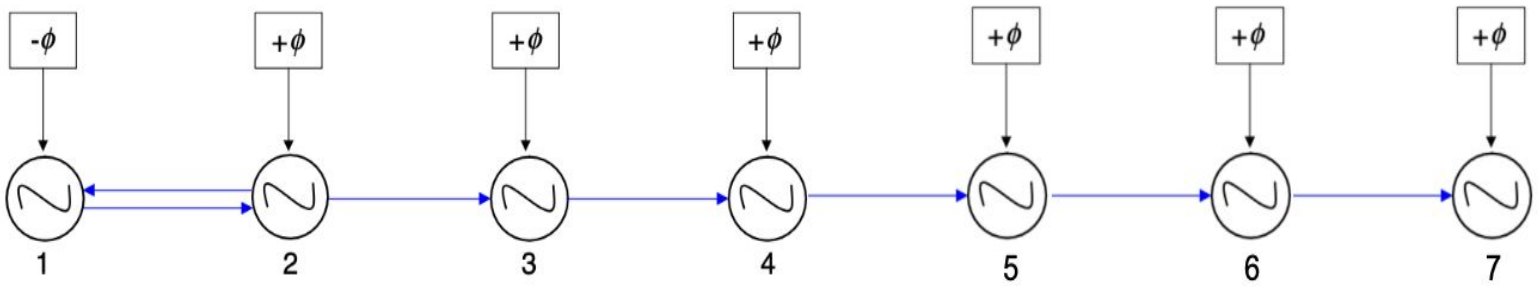
Unidirectional coupling:



Nor & Ma, 2013

Phase Bias Parameter

- $\phi_{ij} = \phi$ for all oscillators
- the sign (±) for ϕ for oscillator 1 → opposite to the other oscillators



Straight Movement Parameters

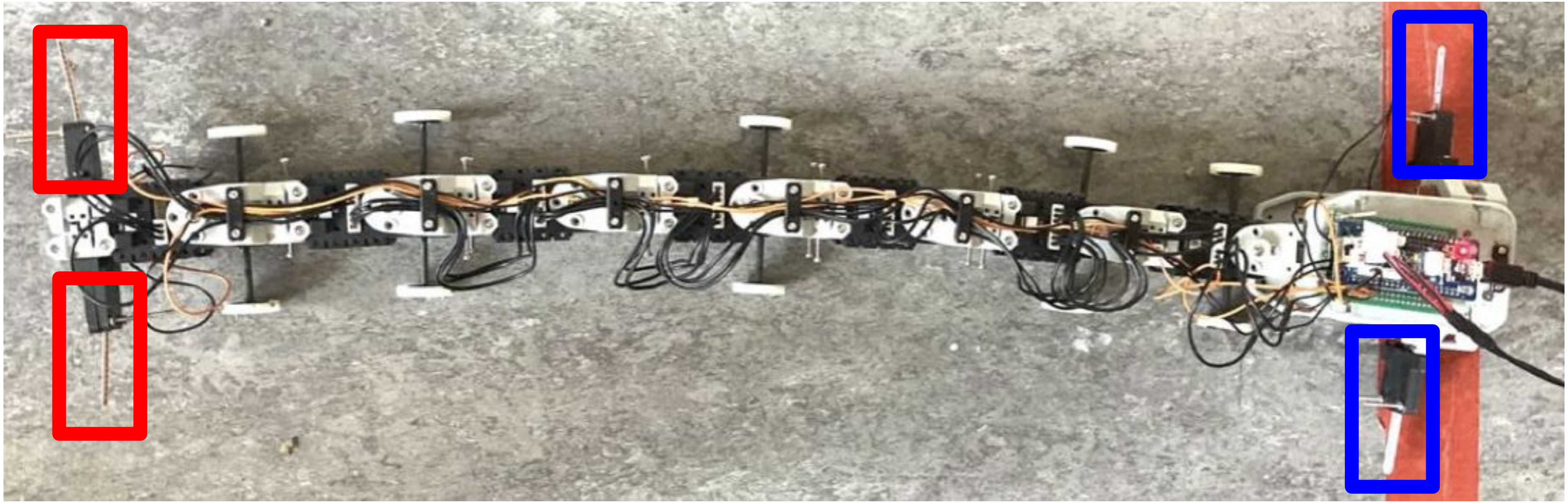
Parameter	***W***	$\boldsymbol{\phi}$	***A***	***trajectory offset***	$\boldsymbol{\tau}$	***v***
Value	10	2π/7	120	511	0.2	0.8

→ same for all oscillators

Changing Behaviour

Back sensors:
Amplitude adjustment

Front sensors:
Left & right movement



- Behavioural change achieved with just 4 sensors
- Sensors at the head & at the tail for better internal representation of environment

Changing Behaviour

***Phase transition-based turning** is not robust in practice (slides 23,24).*
*So, we developed our own approach, based on **offset parameter:***

Left turn: decrease trajectory offset

Parameter	W	ϕ	A	***trajectory offset***	τ	v
Value	10	2π/7	120	max 500 min 451	0.2	0.8

Right turn: increase trajectory offset

Parameter	W	ϕ	A	***trajectory offset***	τ	v
Value	10	2π/7	120	min 522 max 571	0.2	0.8

→ duration: 2 seconds

Changing Behaviour

Phase transition-based approach *is not robust in practice (slides 23,25).*
So, we used an approach based on ***Amplitude and Time-constant parameters:***

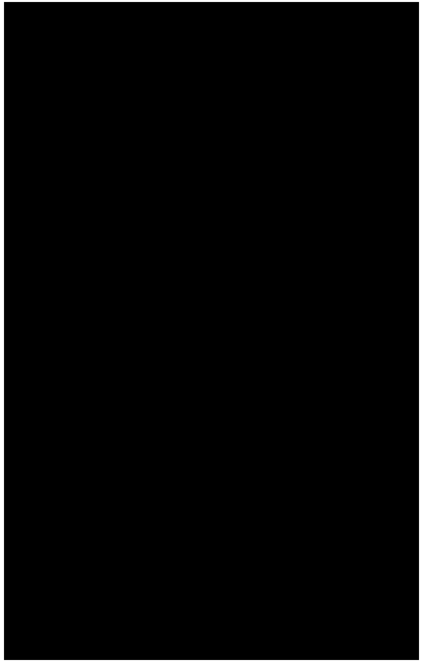
Amplitude adjustment when entering a narrow section:

- Decrease parameter A → decreases speed
- Decrease parameter τ → increases speed

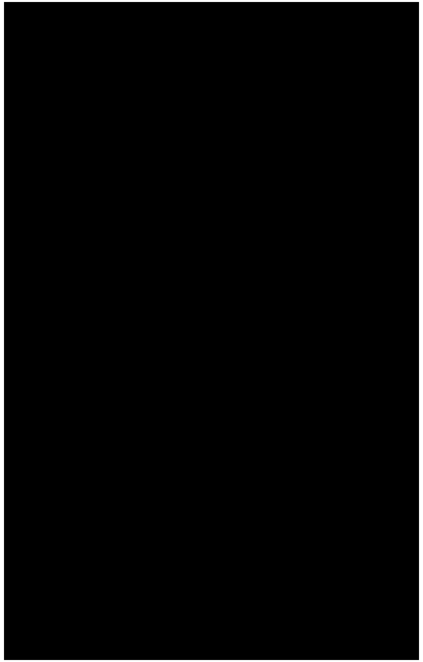
Parameter	***W***	***ϕ***	***A***	***trajectory offset***	$\boldsymbol{\tau}$	***v***
Value	10	2π/7	60	511	0.08	0.8

→ duration: 5 seconds

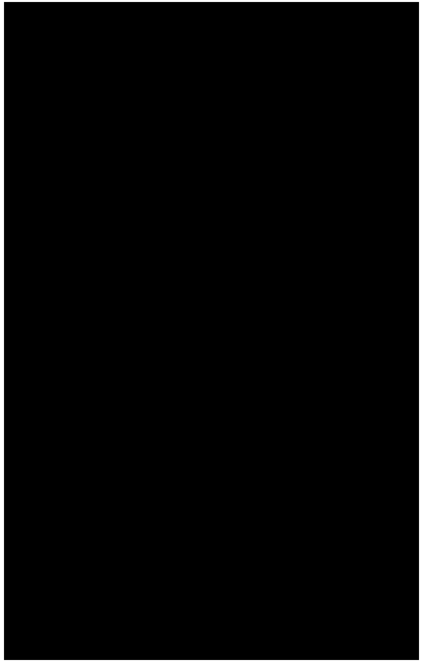
Changing Behaviour



Right turning



Left turning

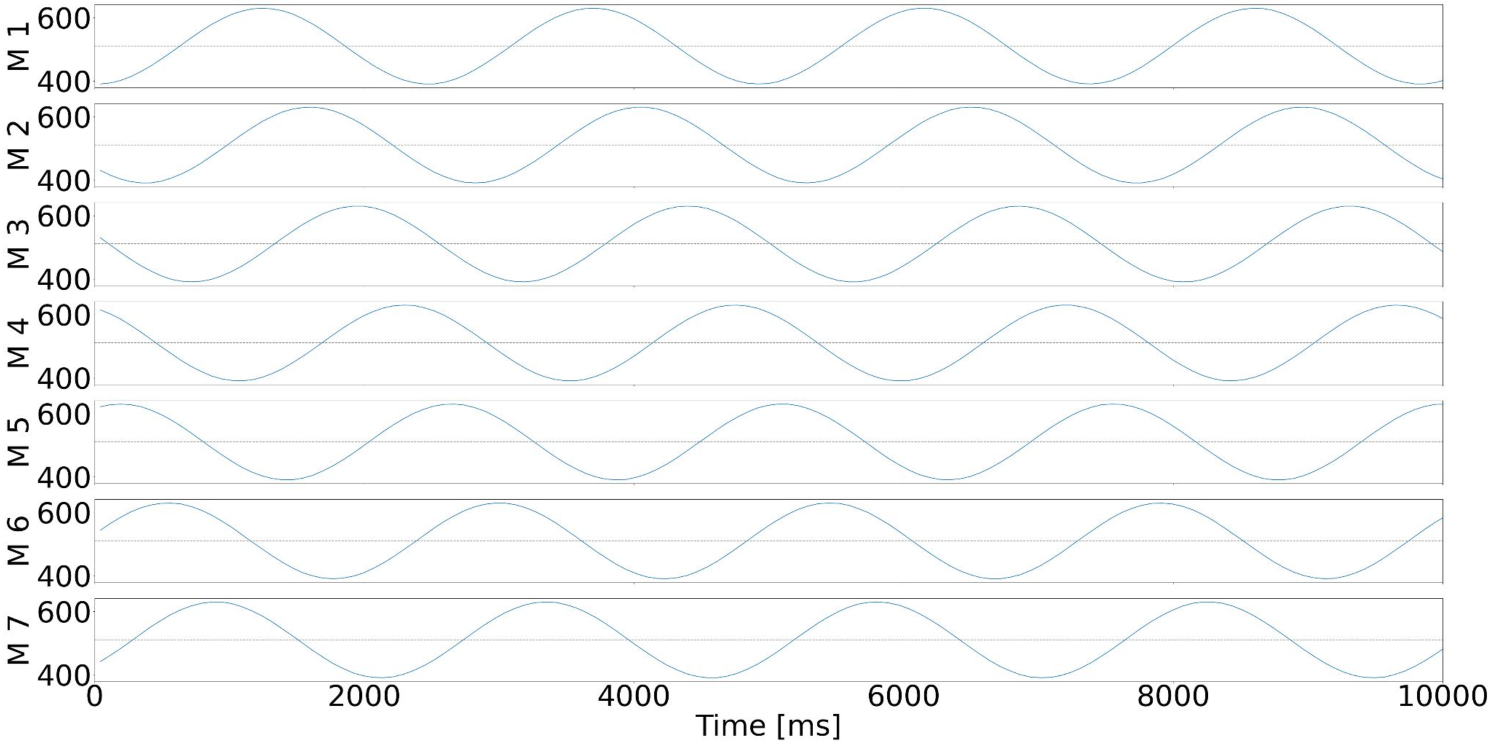


Amplitude change

Evaluations

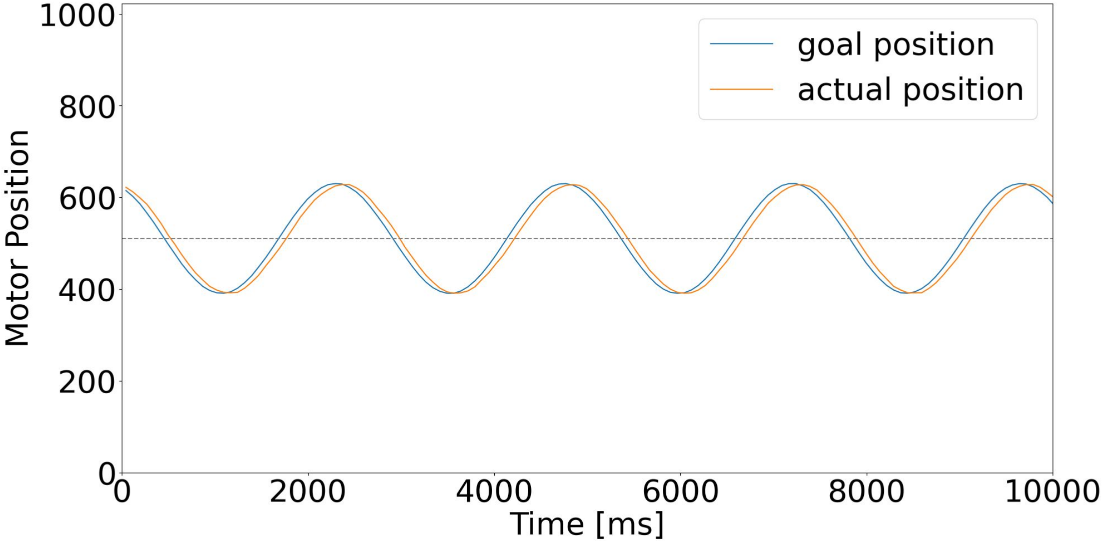
Phase Shift between CPGs

Param.	W	ϕ	A	*offset*	τ	v
Value	10	2π/7	120	511	0.2	0.8



Default Straight Movement

Param.	W	ϕ	A	*offset*	τ	v
Value	10	2π/7	120	511	0.2	0.8



Speed:

0.113 m/s

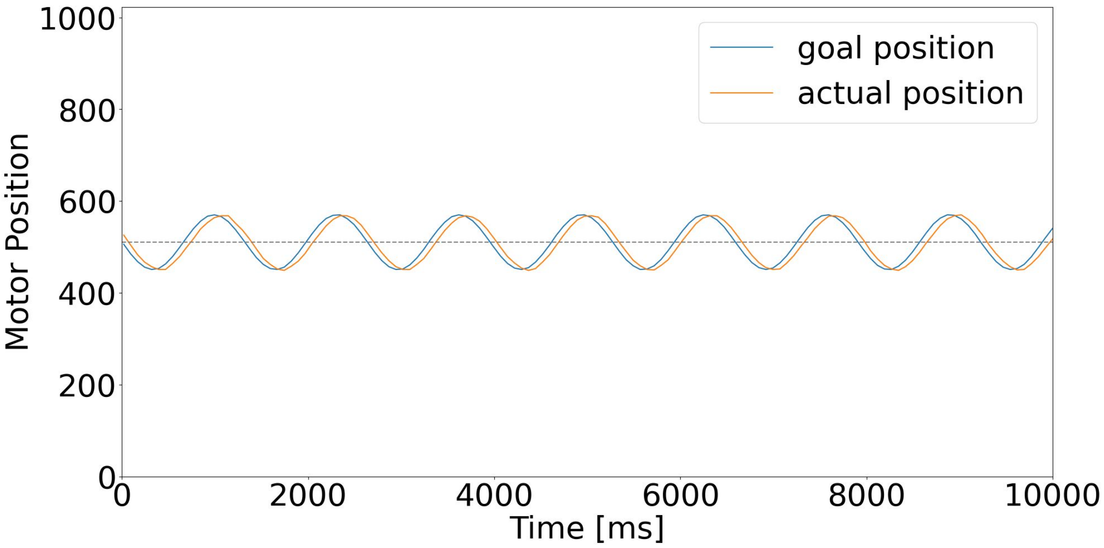
Range of Motion:

Goal Position: 391 - 630

Actual Position: 388 - 629

Narrow Straight Movement

Param.	*W*	*ϕ*	*A*	*offset*	*τ*	*v*
Value	10	2π/7	60	511	0.08	0.8



Speed:

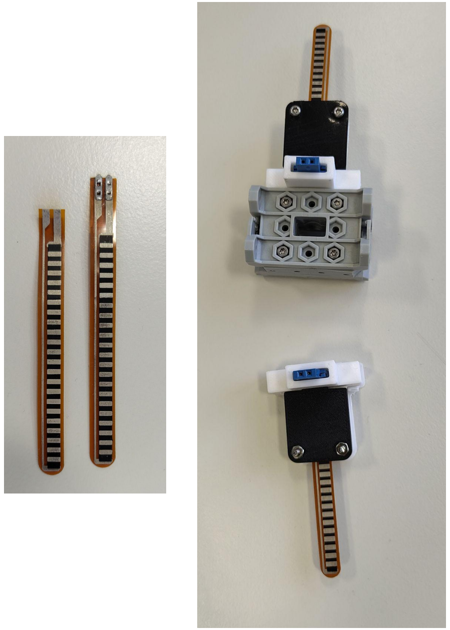
0.124 m/s

Range of Motion:

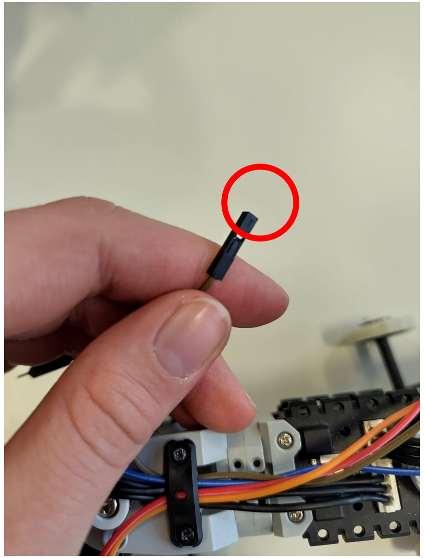
Goal Position: 451 - 570

Actual Position: 448 - 570

Hardware Fixing during the Project

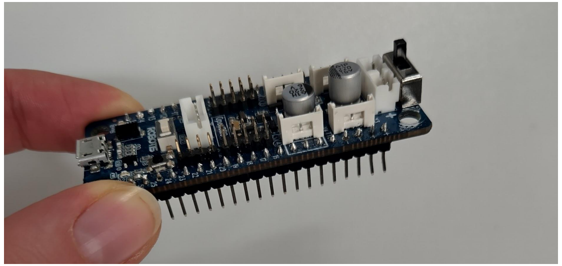


Sensor and sensor holder replacement



Wire replacement

Motor replacement



Sharing a microcontroller

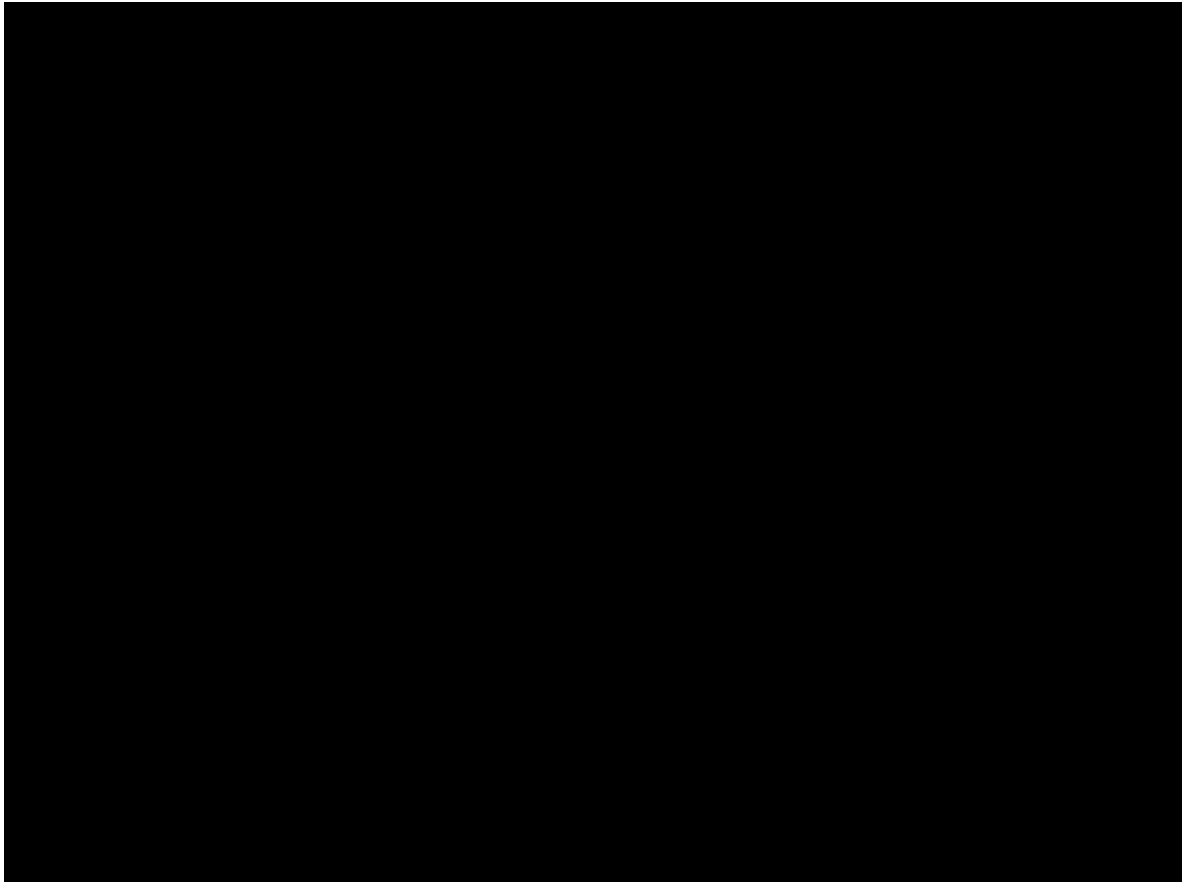
Limitations of Implemented Changing Behaviour

- **Jerky turning** behaviour & **amplitude** change might damage the hardware
  → need for **smoother transitions**

- **A larger number of sensors** would make the robot more robust
  → sooner wide-to-narrow path transition with **sensors in the middle**

- No **parameter tuning** of the phase bias $\boldsymbol{\phi}$
  → might impact the **speed**

Thank you!



References

- Nor, N. M., & Ma, S. (2013). A Simplified CPGs Network with Phase Oscillator Model for Locomotion Control of a Snake-like Robot. *Journal of Intelligent and Robotic Systems*, *75*(1), 71–86. https://doi.org/10.1007/s10846-013-9868-9

- Nor, N. M., & Ma, S. (2014). CPG-based locomotion control of a snake-like robot for obstacle avoidance. *IEEE International Conference on Robotics & Automation (ICRA)*. https://doi.org/10.1109/icra.2014.6906634

Backup Slides

Constraints on Intrinsic Frequency (v) and Coupling Strength (w)

Exemplary for one oscillator θ :

$$\dot{\theta}_3 = 2\pi v + w \sin(\theta_2 - \theta_3 + \phi)$$

Setting the ODE to 0 we obtain:

$$2\pi v + w \sin(\theta_2 - \theta_3 + \phi) = 0$$

Solving for θ_3 we get:

$$\theta_3^{\infty} = \theta_2 + \phi + \sin^{-1}(2\pi v / w)$$

θ_3 will always evolve at constant phase difference of:

$$\theta_2 + \phi + \epsilon, \text{ where } \epsilon = \sin^{-1}(2\pi v / w)$$

for the output of oscillator, to converge to oscillations that are phase locked with a phase difference of ϕ, we need:

$$v << w$$

for ε to become so small that it can be neglected

➔ introduction of parameter τ to control both v and w while keeping the system dynamics

Nor & Ma, 2013

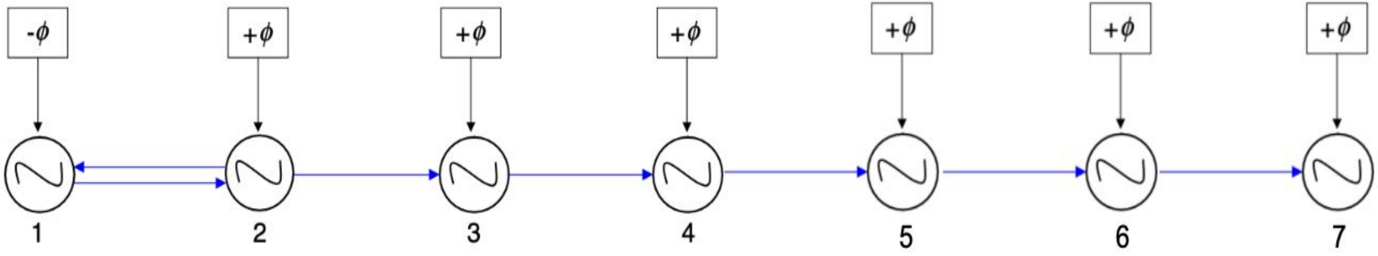
Phase Bias Parameter

- $\phi_{ij} = \phi$ for all oscillators with:

$$\phi = 2\pi N/n$$

N = # of S-shapes ,
n = # of actuated joints from head to tail

- The sign (±) for ϕ for oscillator 1 → opposite to the other oscillators



Modification of Number of S-Shapes (N) via Phase Transition

In order to change number of S-shapes N in robot's locomotion from *N1* to *N2*, phase bias should be changed from ϕ1 to ϕ2. This happens during phase transition time (*t2-t1*), by continuously adjusting bias term as follows :

$$\phi = \phi_1 - \alpha(t_1 - t)$$

$$\alpha = \phi_1(N_2/N_1)(1 - (N_1/N_2))/(t_2 - t_1)$$

Where:

ϕ1 - old phase bias term

ϕ2 - new phase bias term

N1 - old # of S-shapes

N2 - new # of S-shapes

t1 - phase transition start time

t2 - phase transition end time

t - current time

Attempted Implementation of Behaviour Change via Modification of Number of S-Shapes (N) via Phase Transition

We implemented **phase transition via modification of phase bias parameter** as described in *Nor & Ma, 2013*. We applied phase transition strategy to implement:

1. **Turning behavior** by transitioning from N1→N2, where N1>N2
2. **Oscillation amplitude/frequency change in narrow/wide path behavior**, with N1→N2

   where N1>N2 for wide->narrow change

   where N1<N2 for narrow->wide change

For both behaviors, we observed the expected result: change in number of S shapes
However, these **changes were not robust enough** for real-world use *(see next slides)*.

Attempted Implementation of Behaviour Change via Modification of Number of S-Shapes (N) via Phase Transition

Turning behavior:

For turning behavior, N2 must be smaller than N1.
We observed desired pattern change for N2=2 and N1=1 as well as N2=1.5 and N2=1.

Next we needed to find a (t2-t1) parameter which would guarantee that snake makes **less than 0.5 of motion cycle** (otherwise it continues straight motion, with the new number of S-shapes N2).

We used (t2-t1) parameters ranging from 0.1 to 10 seconds:

- For (t2-t1)<2sec, we observed no pattern change
- For (t2-t1)>=2sec, snake makes one movement cycle or more, moving straight (not turning).

One possible reason: the method is not robust enough.
In *Nor & Ma, 2013,* turning behavior via phase transition was simulated,
but no exhaustive experimental results proving robustness of the method were provided.

Attempted Implementation of Behaviour Change via Modification of Number of S-Shapes (N) via Phase Transition

Oscillation amplitude/frequency change in narrow/wide path behavior:

For the above mentioned behavior, depending on environment change, the number of S-shapes should either increase (wide->narrow path) or decrease (narrow->wide path). After this change, straight movement should proceed.

We observed expected pattern changes for N1=1 and N2=2, N1=1 and N2=3, N1=1 and N2=1.5, N1=2 and N=1, N2=1 and N2=1, N1=1.5 and N2=1.

However, we observed, that during transitions N1->N2 **where N2>N1**, the snake develops N2>4 some time after phase transition. The number of S-shapes is too large for effective straight movement, so the **snake does not move effectively** (even though motors are turning).

One possible reason: flaws in our implementation.

Even though the approach worked well for N1>N2, it failed for N1<N2, so we could not use it.

Parameter Tuning

Parameter	*W*	*ϕ*	*A*	*offset*	*τ*	*v*	*speed*
Nor & Ma, 2013	10	2π/7	1	511*	0.2*	0.1	0 m/s
Measurement 2	10	2π/7	120	511	0.2	0.1	0.013 m/s
Measurement 3	10	2π/7	120	511	0.08	0.1	0.026 m/s
Measurement 4	10	2π/7	120	511	0.08	0.8	0.232 m/s
Chosen Default	10	2π/7	120	511	0.2	0.8	0.113 m/s
Measurement 6	10	2π/7	60	511	0.2	0.8	0.054 m/s
Chosen Narrow	10	2π/7	60	511	0.08	0.8	0.124 m/s

no movement
→ increase A

very slow
→ decrease τ **

still slow
→ increase v

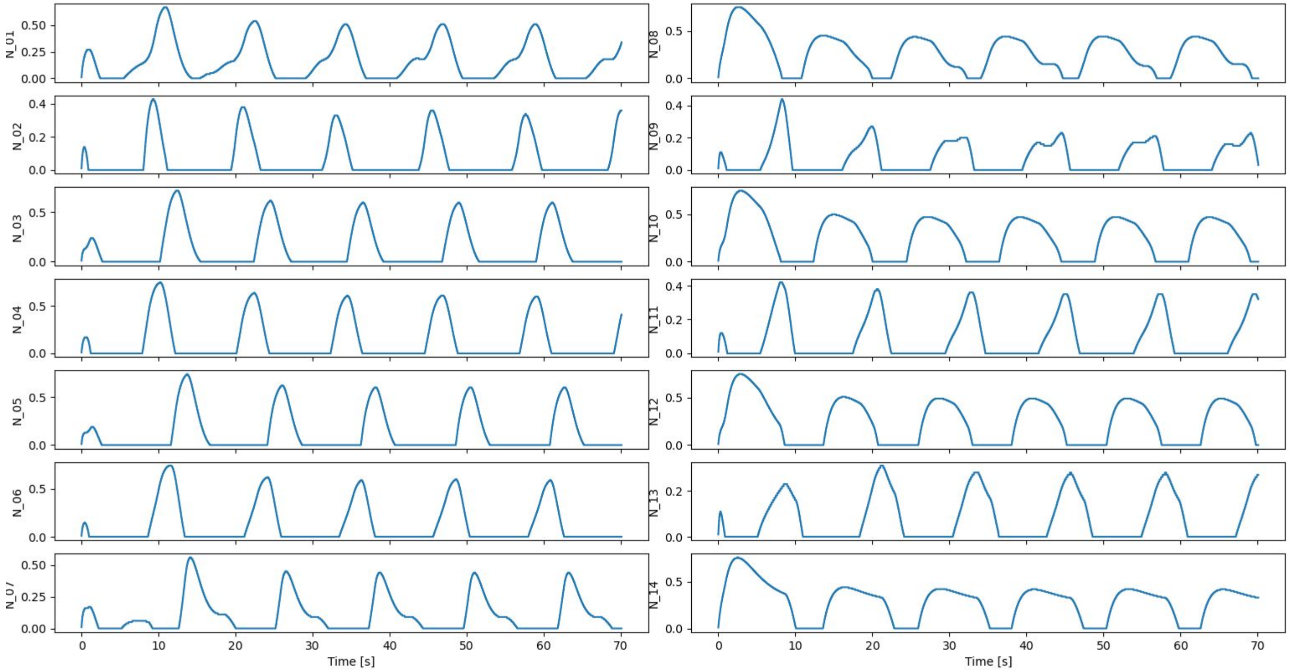
violent speeding
→ decrease τ

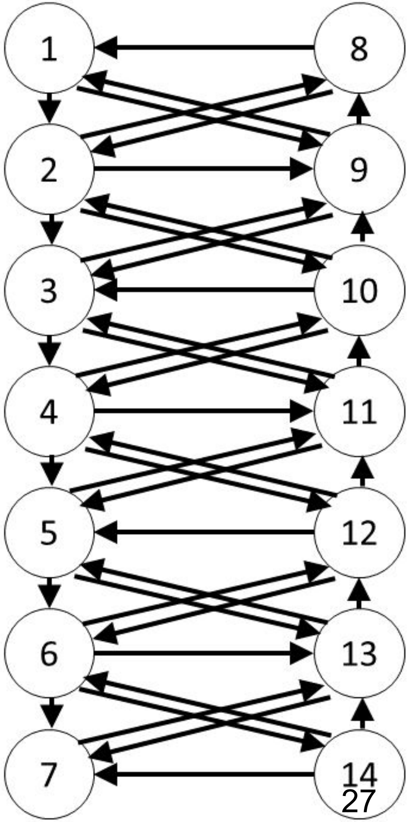
explore narrow movement
→ decrease A

too slow
→ decrease τ

* not specified in paper, 1/20 ≥ τ ≤ 1, ** for relationship v, w, τ see slide 20

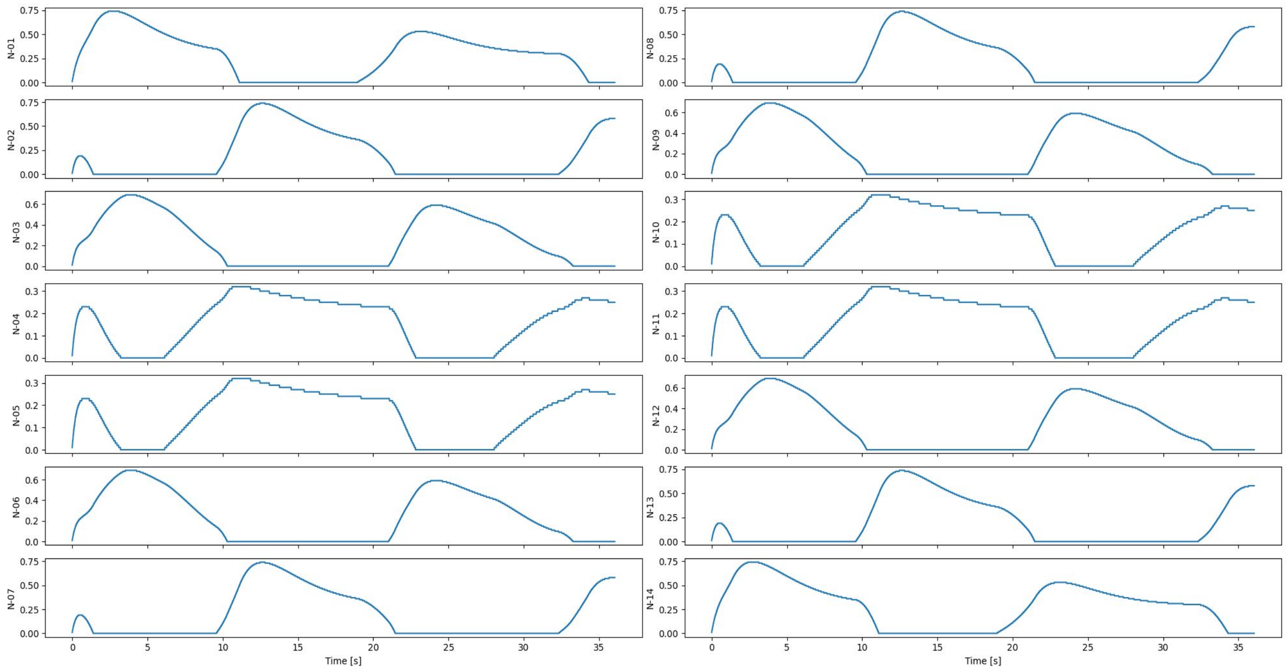
Matsuoka Model Configurations Testing

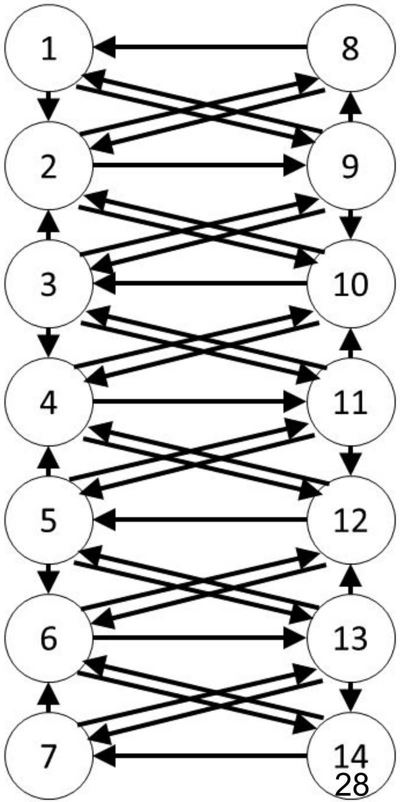




→ no horizontal mutual inhibition, phase difference too large, varying shapes

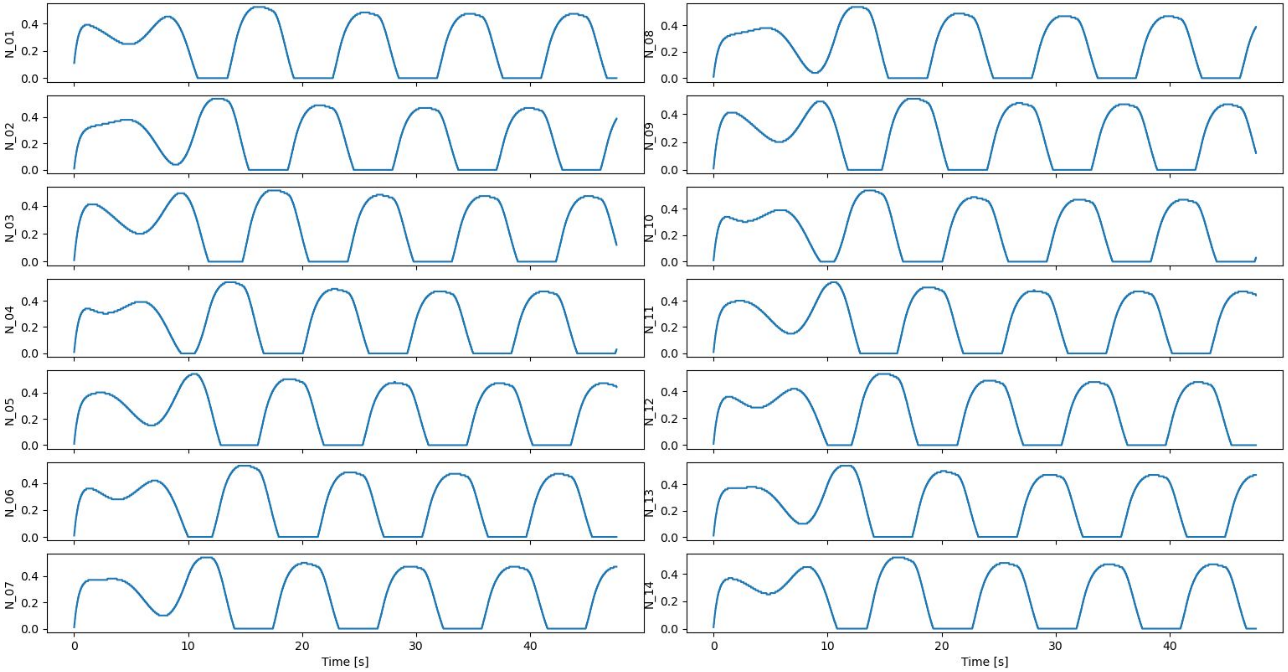
Matsuoka Model Configurations Testing

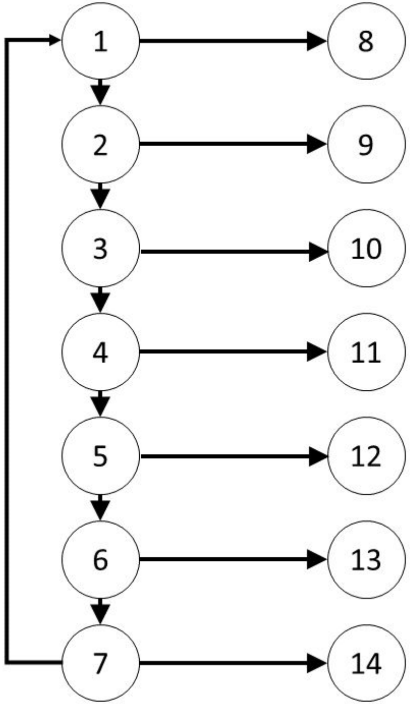




→ period and phase difference too large, varying shapes

Matsuoka Model Configurations Testing





→ phase difference too large