Motivation of this Research

◆ Snakes perform many kinds of movement that are adaptable to a given environment by changing locomotion modes

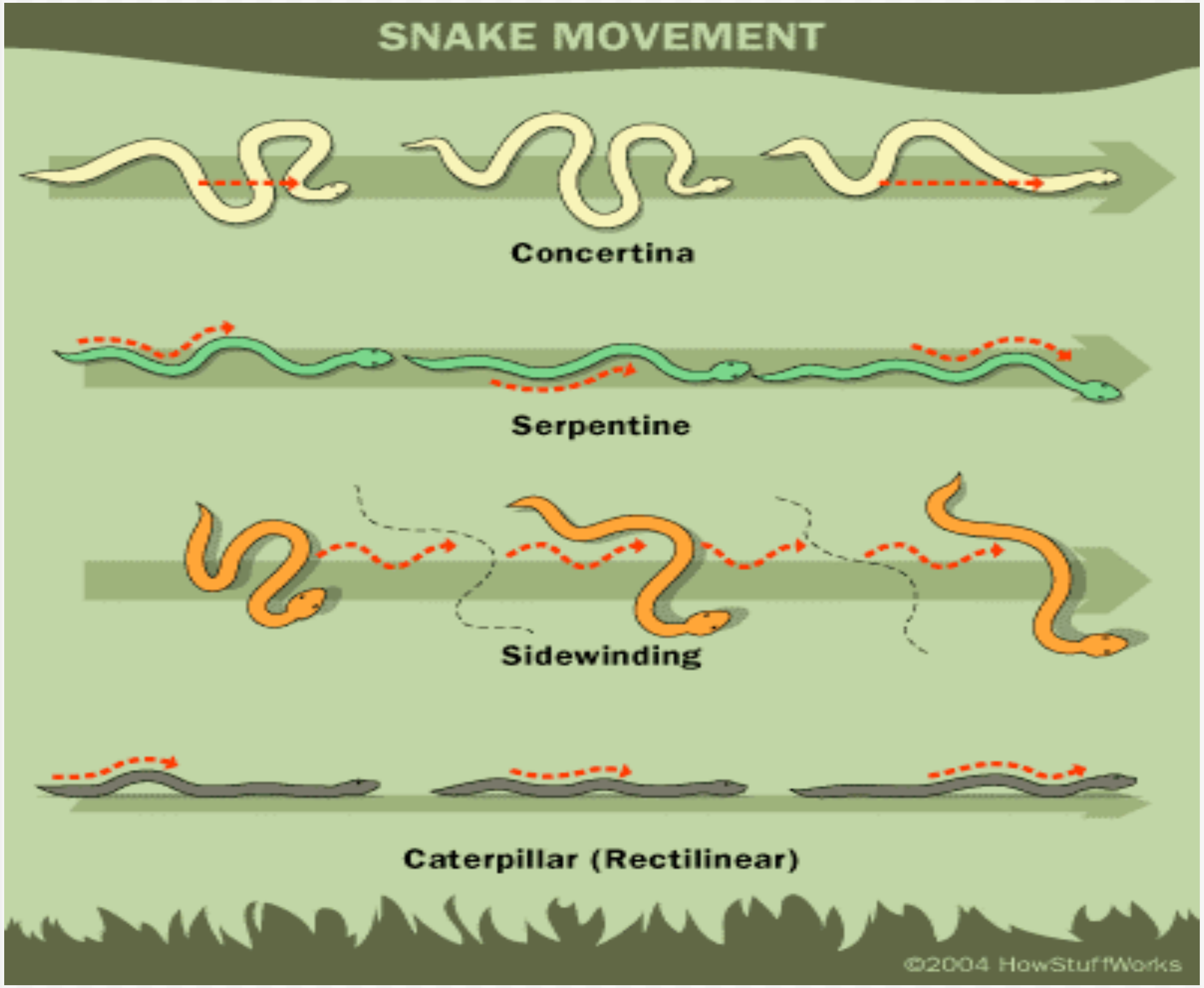
Move on soft ground

Move across branches

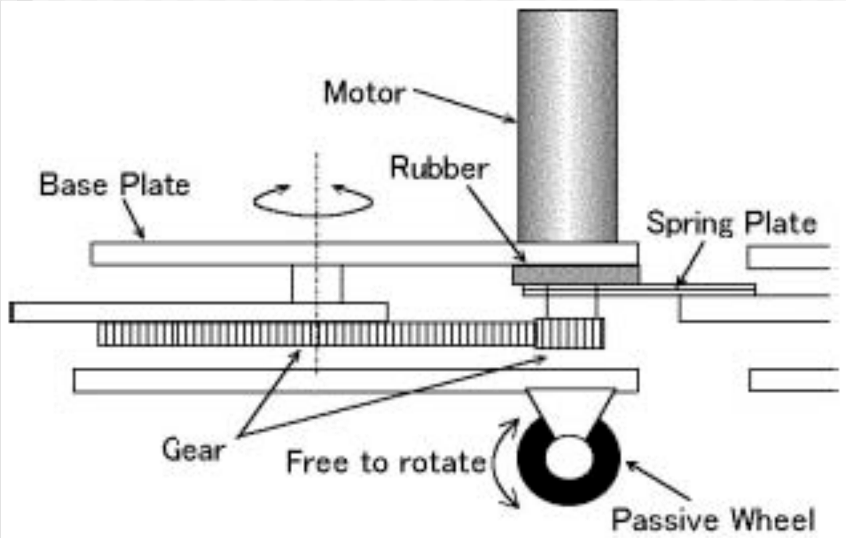
Snake robots are potentially superior for operations in highly constrained and unusual environments encountered in applications:

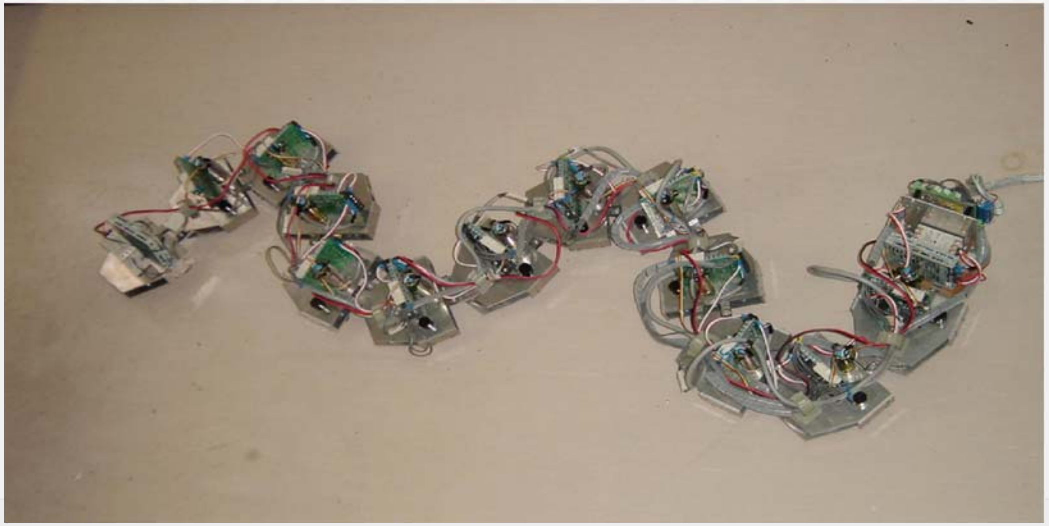
- Inspection of nuclear reactor cores and chemical sampling of buried toxic waste
- Space applications such as exploration of planetary surfaces and planet sample return mission
- Rescue task like searching of victims in the debris after a disaster
- Underwater applications such as ocean exploration and oil field service

Motion Examples of Snakes

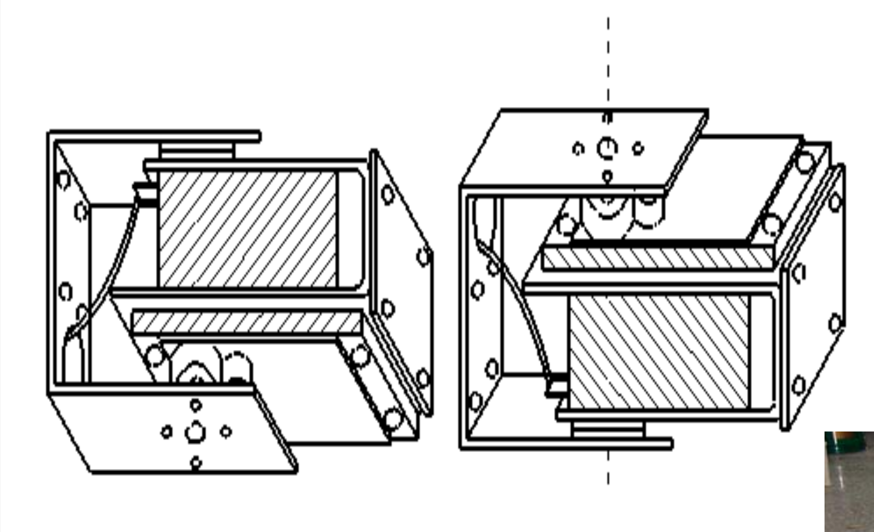


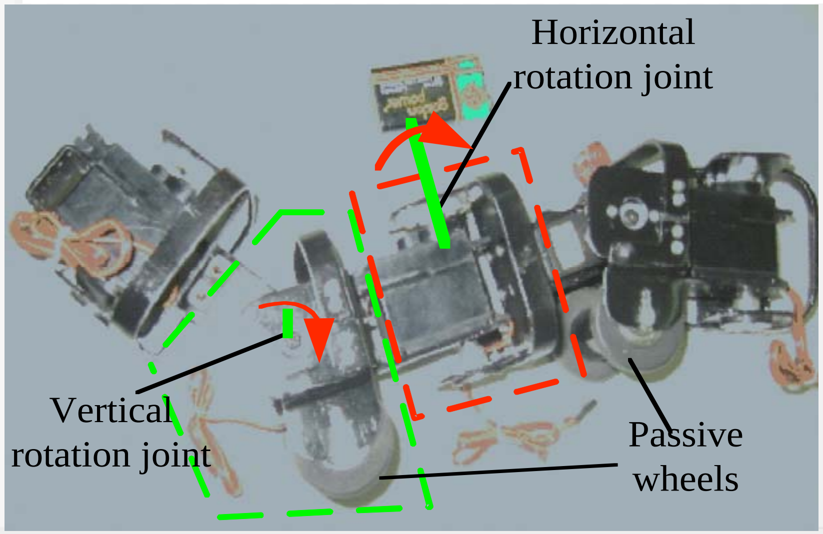
Design of 2D Snake Robot
(1 DOF Joint)



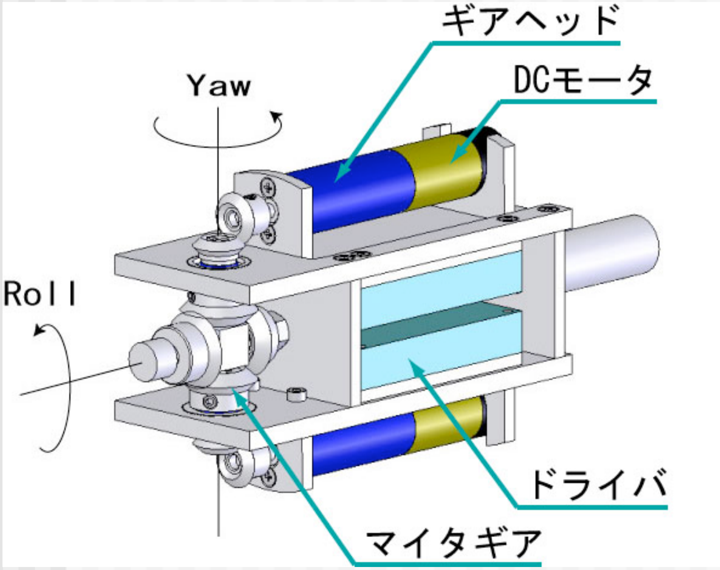


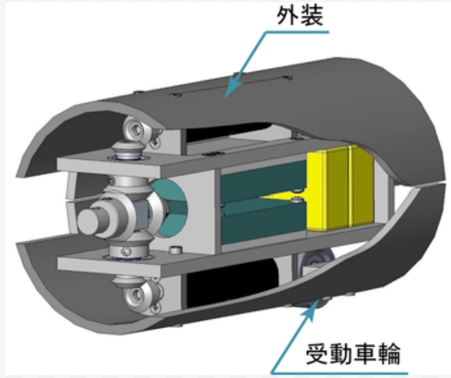
Design of 3D Snake Robot
(1 DOF Joint)

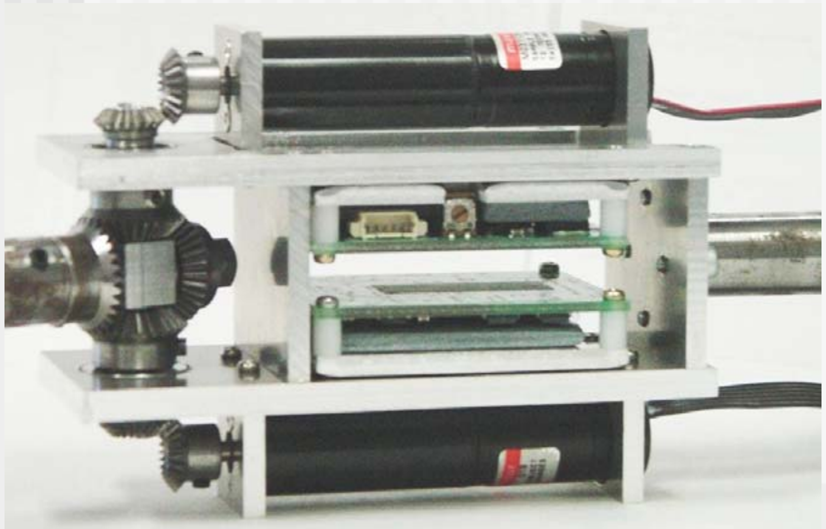




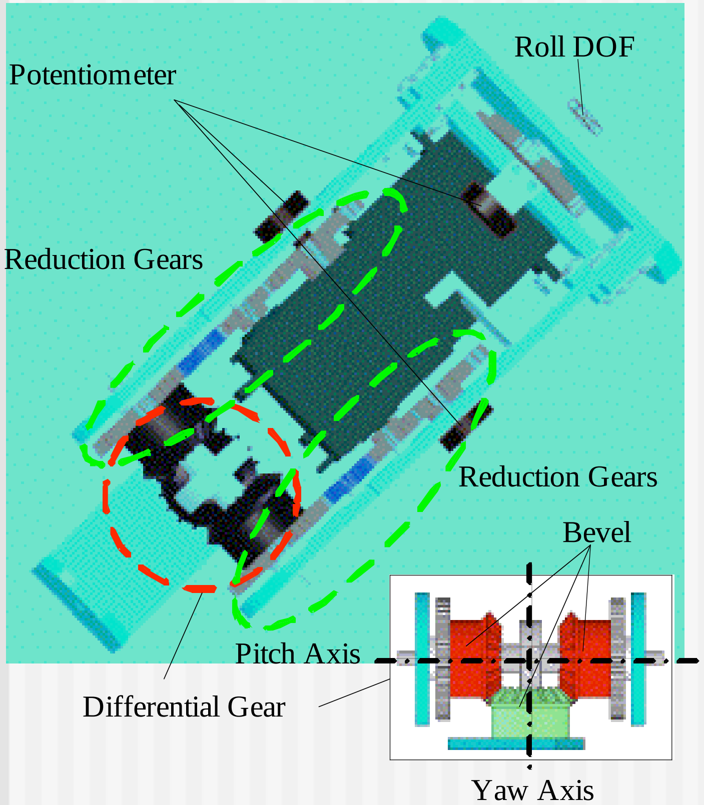
Design of 3D Snake Robot
(2 DOF Joint)

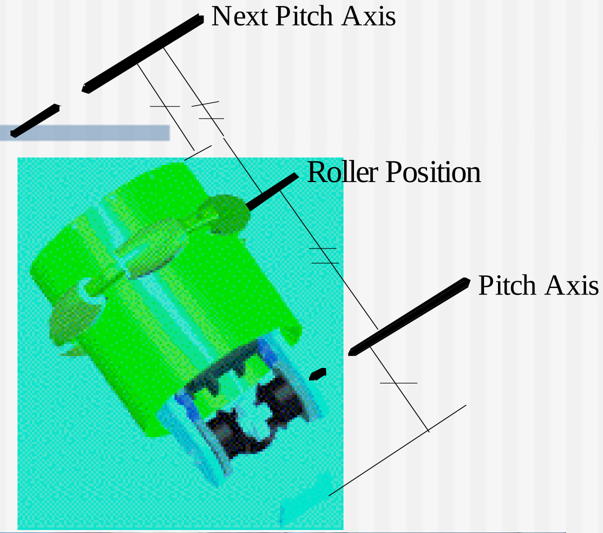




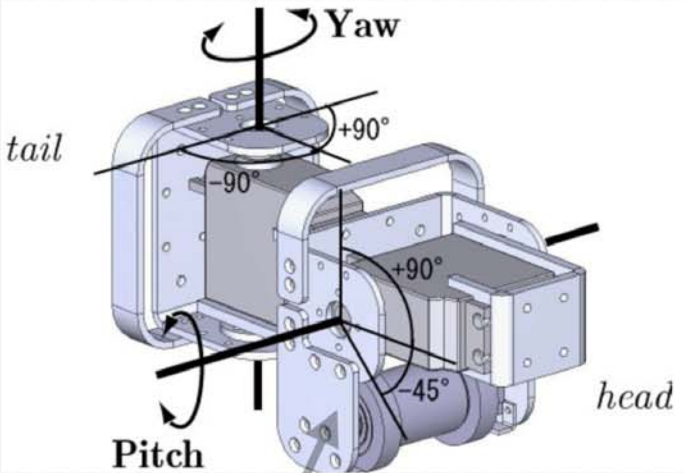


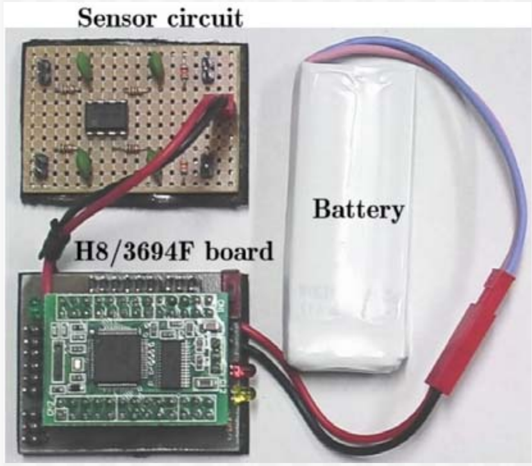
Design of 3D Snake Robot
(3 DOF Joint)

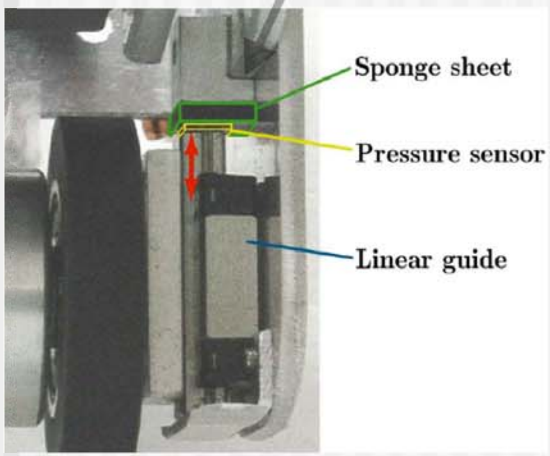


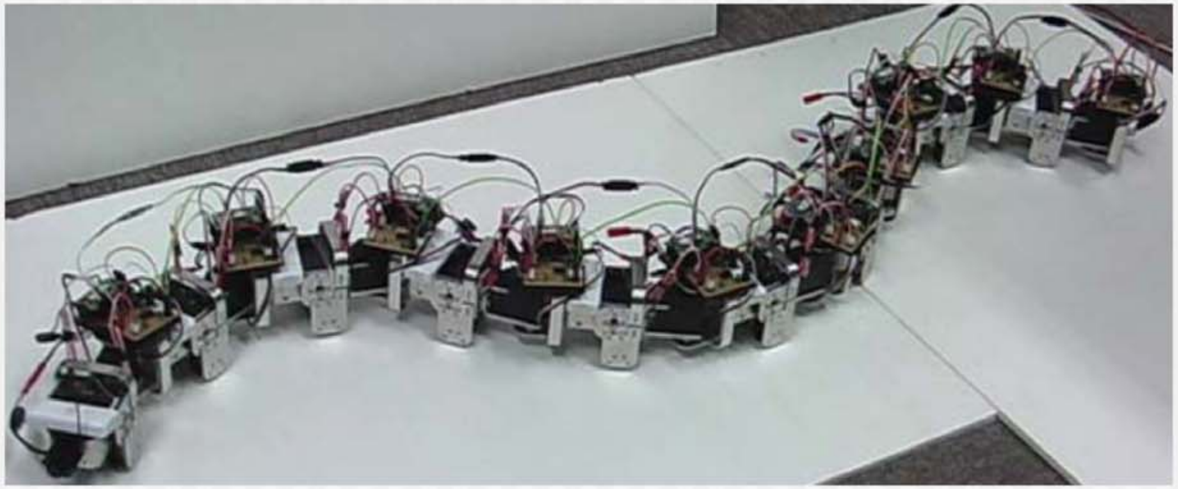


Design of 3D Snake Robot for Environmental Adaptation





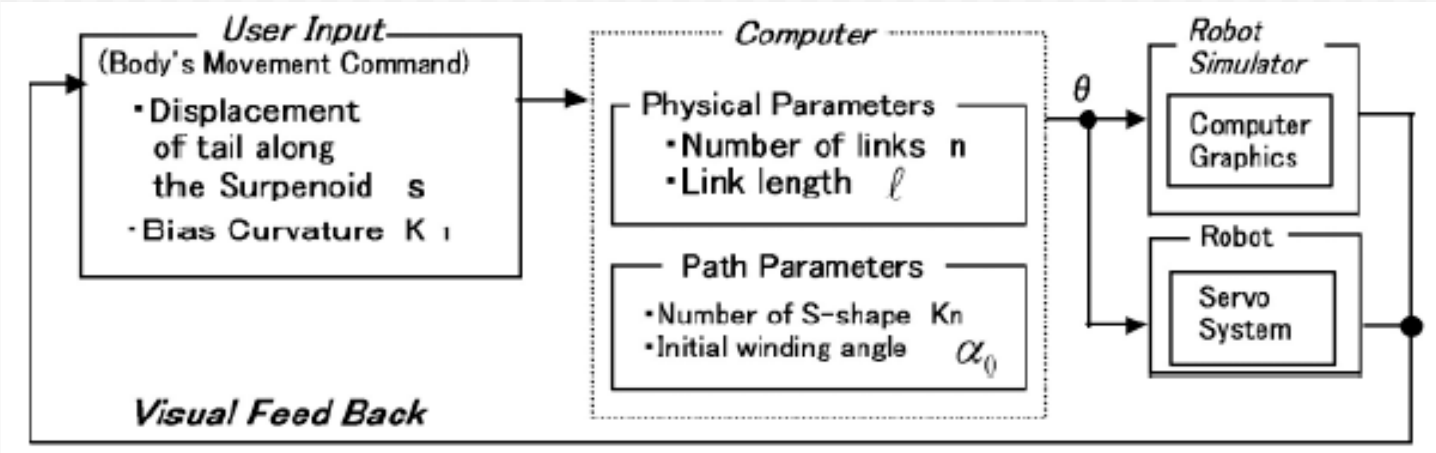




Control of Snake Robots

- Analytical model of body dynamics for known environment
- Rhythmic motion generated by neural oscillator networks

Control System of 2D Motion of Snake Robots



Serpentine :

$$\begin{bmatrix} \theta_i^y(s) \\ \theta_i^p(s) \end{bmatrix} = \begin{bmatrix} -2\alpha_0^y \sin\left(\frac{K_n\pi}{n}\right) \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i\right) \\ 0 \end{bmatrix}$$

Sinusoidal :

$$\begin{bmatrix} \theta_i^y(s) \\ \theta_i^p(s) \end{bmatrix} = \begin{bmatrix} 0 \\ -2\alpha_0^p \sin\left(\frac{K_n\pi}{n}\right) \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i\right) \end{bmatrix}$$

Control System of 3D Motion of Snake Robots

Sinus-lifting :

Wavelength: Yaw:Pitch=1:2	Phase Difference: $-\pi/2$

$$\begin{bmatrix} \theta_i^y(s) \\ \theta_i^p(s) \end{bmatrix} = -2 \begin{bmatrix} \alpha_0^y \sin\left(\frac{K_n\pi}{n}\right) \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i\right) \\ \alpha_0^p \sin\left(\frac{2K_n\pi}{n}\right) \sin\left(\frac{4K_n\pi}{L}s + \frac{4K_n\pi}{n}i - \frac{\pi}{2}\right) \end{bmatrix}$$

Sidewinding :

Wavelength: Yaw:Pitch=1:1	Phase Difference: $-\pi$

$$\begin{bmatrix} \theta_i^y(s) \\ \theta_i^p(s) \end{bmatrix} = -2\sin\left(\frac{K_n\pi}{n}\right) \begin{bmatrix} \alpha_0^y \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i\right) \\ \alpha_0^p \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i - \pi\right) \end{bmatrix}$$

Analysis of Creeping Locomotion

- Analysis of snake creeping locomotion

  Elucidated the standard creeping movement form of a snake through analyzing physiologically

- Analysis of creeping locomotion of snake-like robot

  The number of $S-$ shape does not give large influence on the performance, but the initial winding angle largely does

- Analysis of creeping locomotion of snake-like robot on slopes

  ▲The case that, the number of $S-$ shape = 2, is better used for our 12-link snake-like robot

  ▲ The unsymmetrical body shape is better used to improve the robot's performance on the slope

Control of Snake Robots

- Analytical model of body dynamics for known environment

- Rhythmic motion generated by neural oscillator networks

Rhythmic motion generated by neural oscillator networks

Biologically:

Rhythmic locomotion of animals:

Generated by neural oscillator networks located in spinal cord

- Construction of models

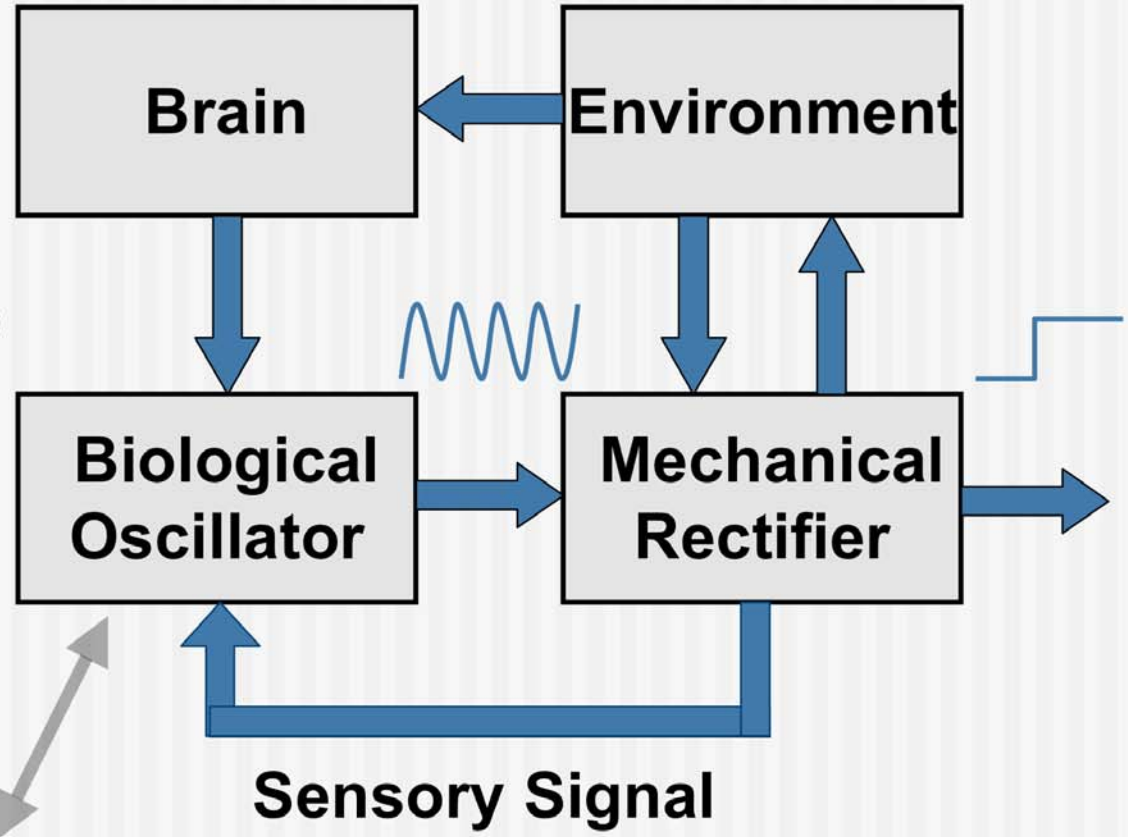
Engineering:

Biologically-inspired Robots:

Needs for Adaptive Controllers for Rhythmic Motion

- Application of neural oscillator network model

Biological Control for Locomotion



Neural Oscillator network

Snake-like Robots

Special Features:

- Many units connected in series
- Interact with environments only through friction
- Rhythmic locomotion

⬇

- Difficulty in calculating body dynamics
  (large DOF, complex interaction with environment)
- Difficulty to generate purposive motion
  in dynamic or unknown environment

⬇

Decentralized Control by Neural Oscillator Network

Snake-like Robots

Analytical model of body dynamics
for known environment

Computational complexity, lower adaptability

Rhythmic motion generated by
neural Oscillator networks

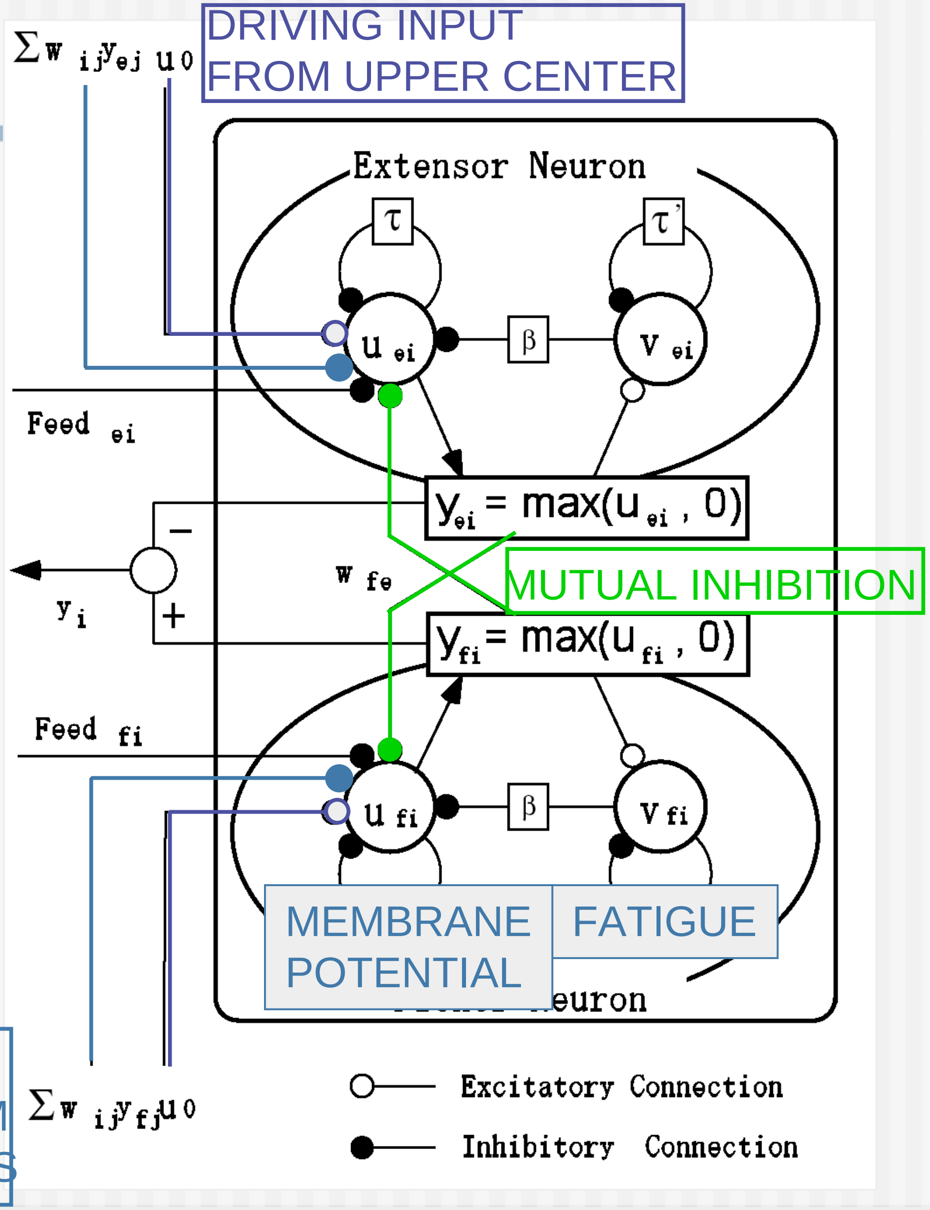
Lower computation, fast adaptation

Matsuoka's Neural Model

Characteristics:

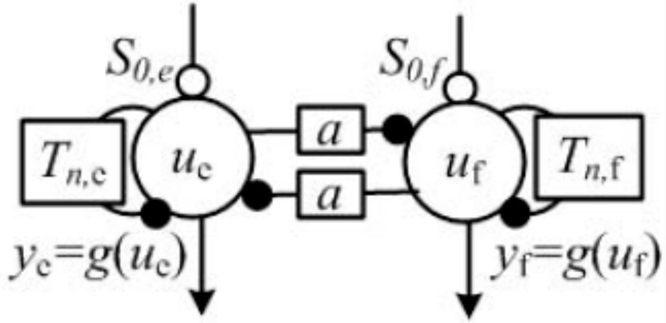
- Mutually inhibiting neurons
- Fatigue effect in each neuron

$$
\begin{aligned}
\tau \dot{u}_{\{e,f\}i} &= -u_{\{f,e\}i} + w_{fe} y_{\{f,e\}i} \\
&\quad -\beta v_{\{e,f\}i} + u_{0,\{e,f\}i} \\
&\quad + Feed_{\{e,f\}i} + \sum_{j=1}^{n} w_{ij} y_{\{e,f\}j} \\
y_{\{e,f\}i} &= \max(0, u_{\{e,f\}i}) \\
\tau' \dot{v}_{\{e,f\}i} &= -v_{\{e,f\}i} + y_{\{e,f\}i} \\
y_i &= y_{fi} - y_{ei}
\end{aligned}
$$

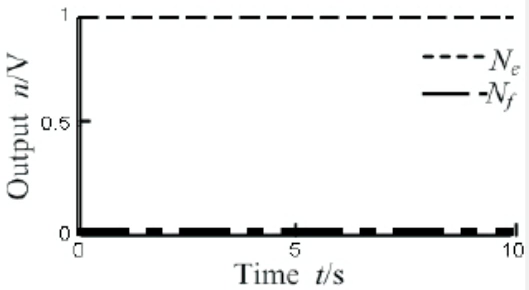


CPG model by Matsuoka

Properties of Mutual Inhibitory CPG Model (without FATIGUE)



(a) Structure



(b) Output of the neurons

$$T_{n,e}\dot{u}_e + u_e = s_{0,e} - ag(u_f)$$
$$T_{n,f}\dot{u}_f + u_f = s_{0,f} - ag(u_e)$$
$$y_{\{e,f\}} = g(u_{\{e,f\}}), g(u_{\{e,f\}}) = max(0, u_{\{e,f\}})$$
$$c\_out = y_e - y_f$$

STATIONARY SOLUTIONS

Conditions	Stationary states(u_1 , u_2)	Stability
$s_{0,1}<0$, $s_{0,2}>0$	$(s_{0,1}-as_{0,2},\ s_{0,2})$	Stable
$s_{0,1}>0$, $s_{0,2}<0$	$(s_{0,1},\ s_{0,2}-as_{0,1})$	Stable
$s_{0,1}<0$, $s_{0,2}<0$	$(s_{0,1},\ s_{0,2})$	Stable

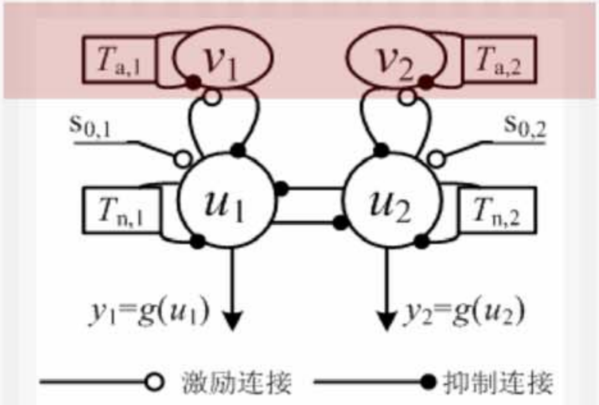
STATIONARY SOLUTIONS

Conditions	Stationary states(u_1 , u_2)	Stability
a<s	$(\frac{s_{0,1}-as_{0,2}}{1-a^2}, \frac{s_{0,2}-as_{0,1}}{1-a^2})$	Stable
s<a<S, $s_1>s_2$	$(s_1, s_2-as^*_1)$	Stable
s<a<S, $s_1<s_2$	$(s_1-as^*_2, s_2)$	Stable
a>S	$(\frac{s_{0,1}-as_{0,2}}{1-a^2}, \frac{s_{0,2}-as_{0,1}}{1-a^2})$	Unstable
	$(s_1, s_2-as^*_1)$	Stable
	$(s_1-as^*_2, s_2)$	Stable

$S=max\{s_e, 1/s_f, s_f/s_e\}$

The Mutual Inhibitory CPG Model without "Fatigue" never yield any oscillatory behavior

Properties of Mutual Inhibitory CPG Model (with FATIGUE)



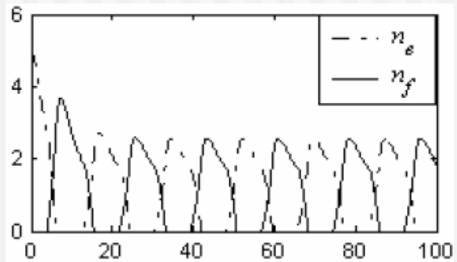
$$T_{n,1}\dot{u}_1 + u_1 = s_{0,1} - a\,g(u_2) - \beta v_1$$

$$T_{a,1}\dot{v}_1 + v_1 = y_1$$

$$T_{n,2}\dot{u}_2 + u_2 = s_{0,2} - a\,g(u_1) - \beta v_2$$

$$T_{a,2}\dot{v}_2 + v_2 = y_2$$

$$y_i = g(u_i) \qquad g(u_i) = \max(0, u_i) \qquad i = 1, 2$$



(a)
Output of neurons

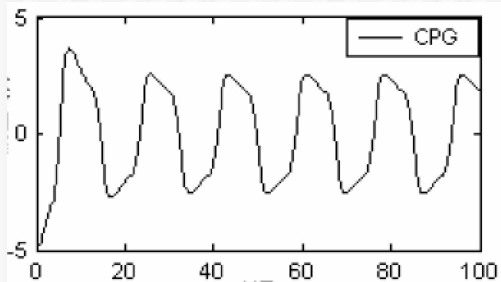
Theorm 1: No stable stationary solution, if and only if

$$a' < s$$
$$a > 1 + \frac{T_n}{T_a}$$

where

$a' = a/(1+\beta)$, $s = \min(s_{0,1}/s_{0,2}, s_{0,1}/s_{0,2})$

Theorm 2: Any solutions are bounded for t > 0 while $a \geq 0$.



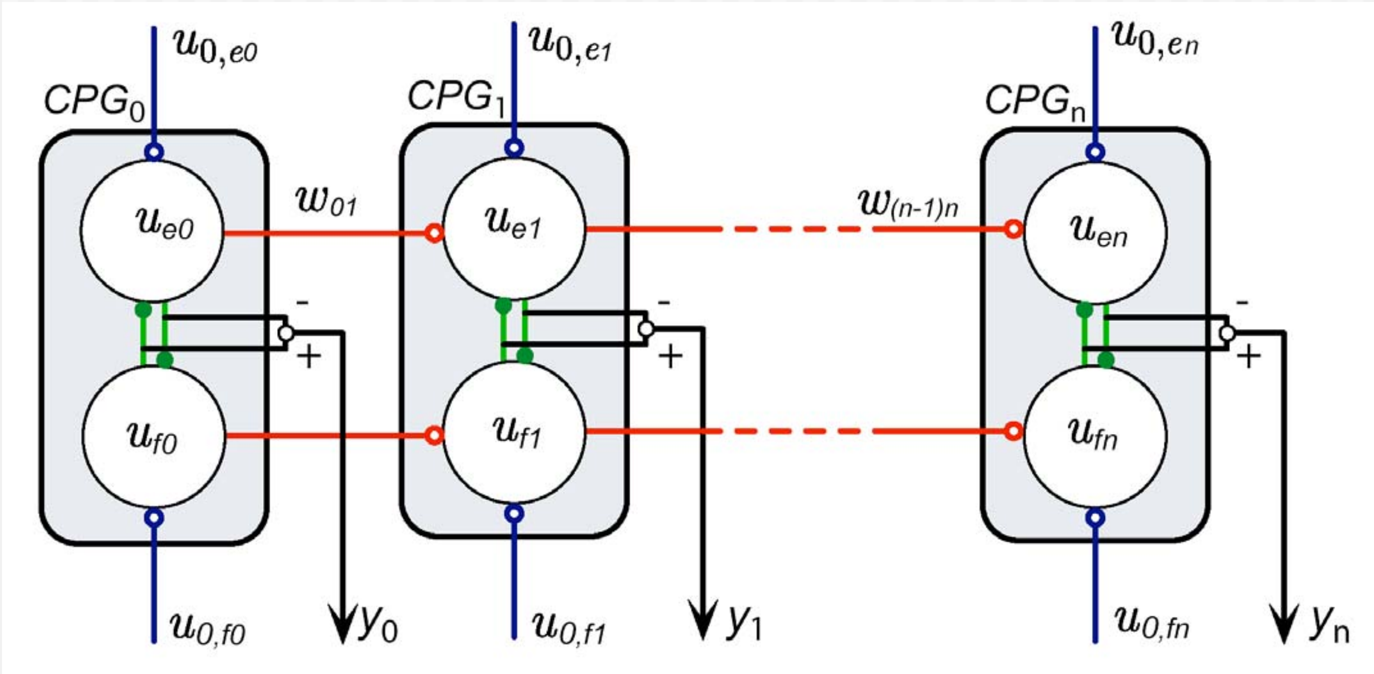
(b)
Output of CPG

The Mutual Inhibitory CPG Model with "Fatigue" yield oscillatory behavior

Network Structure

One-way excitatory connection from head to tail ➡ Propagation of undulation with specific phase difference

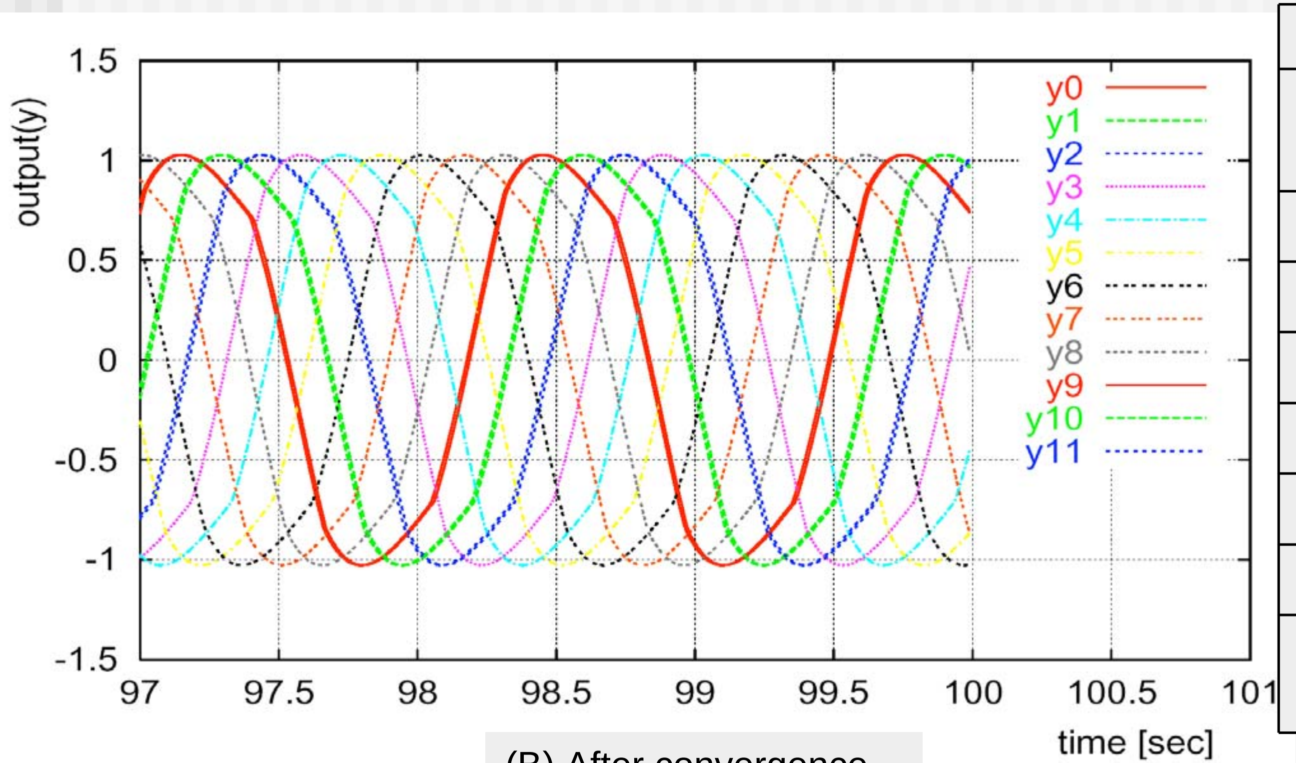
HEAD



TAIL

CPG network

Neural Oscillator Simulation



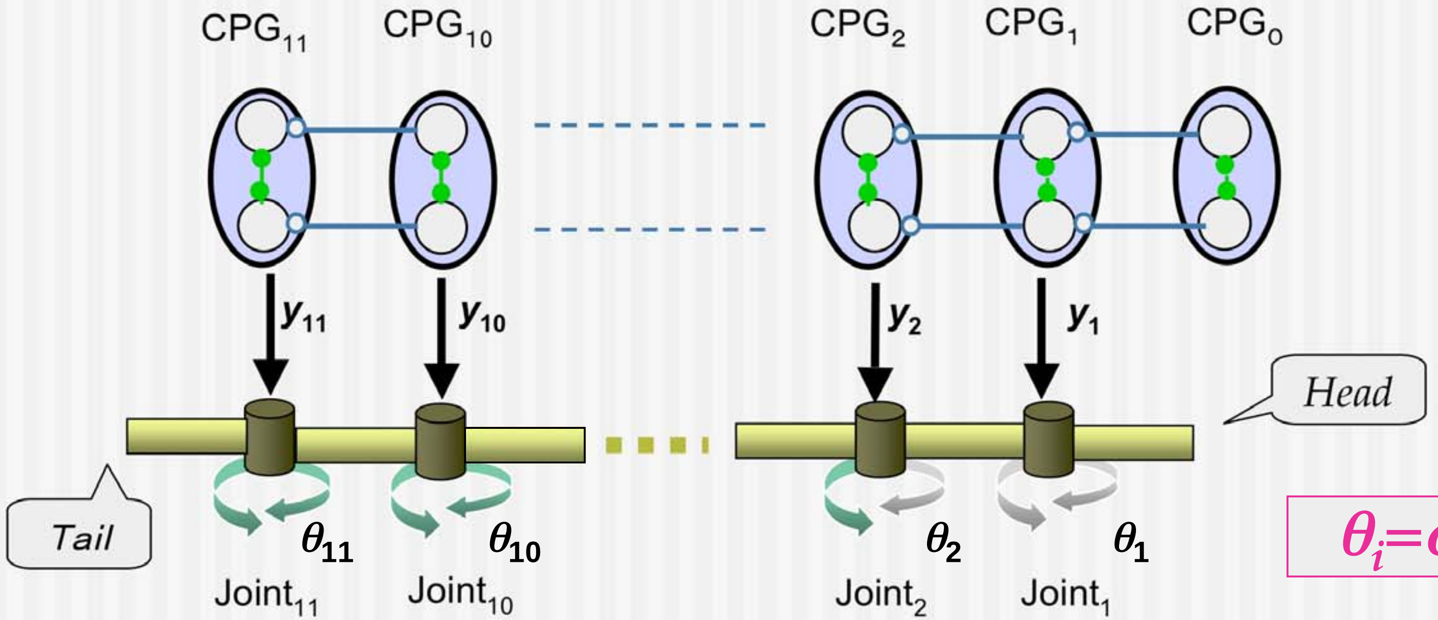
(B) After convergence

Parameters	Value
$u_{0\{e,f\}0}(u_{0\{e,f\}i})$ $(i=1\ldots11)$	3.55(3.0)
t	0.2
t'	1.0
b	5.0
w_{fe}	-1.2
w_{ji}	0.2
cycle[s]	1.3
Phase Difference	45°

(set by trial-and-error)

After 35 seconds, all CPGs oscillate with 1.3[s] cycle with 45[deg] phase difference

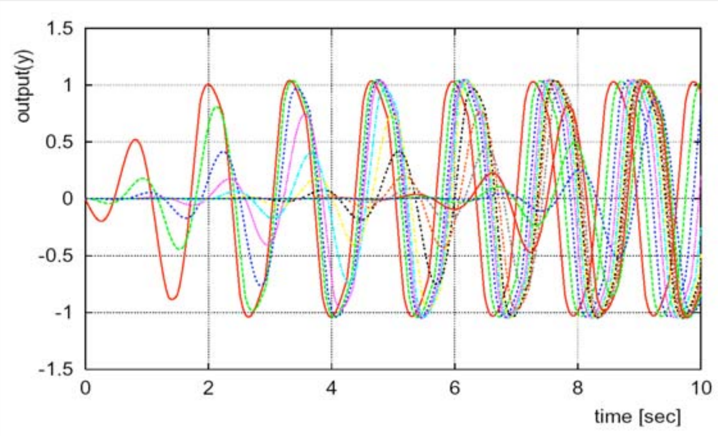
Implementation to Snake Robot



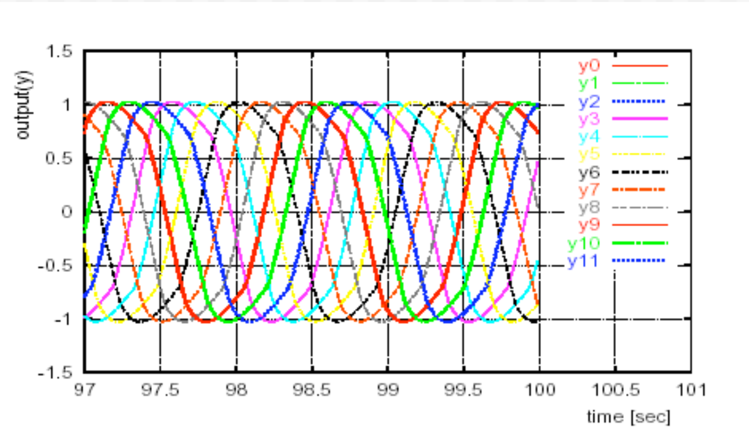
$$\theta_i = \alpha\, y_i$$

- Output of CPGs are input to joint as angle
- CPG_0 is used as a driving input to the network

Simulation Result



Initial stage

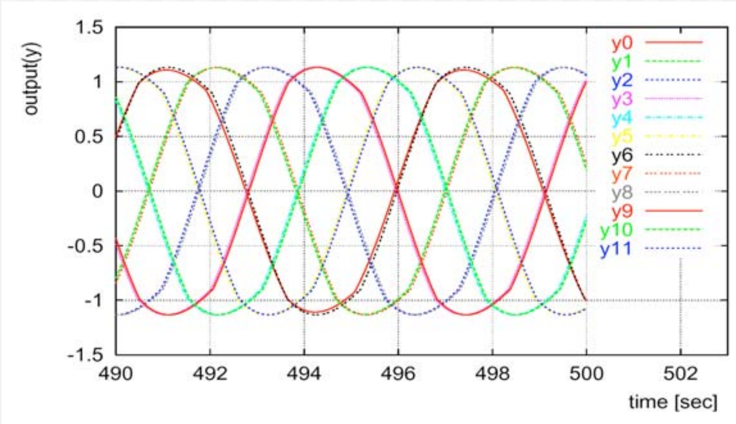


After convergence

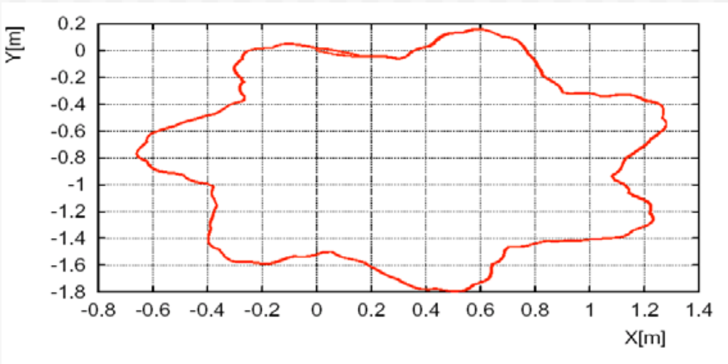
Parameters	Value
$u_{0\{e,f\}0}(u_{0\{e,f\}i})$ $(i=1 \ldots 12)$	3.55(3.0)
t	0.2
t'	1.0
b	5.0
w_{fe}	-1.2
w_{ji}	0.2
Cycle [s]	1.3
Phase difference [deg]	45°

(set by trial-and-error)

Simulation Result



Steady stage



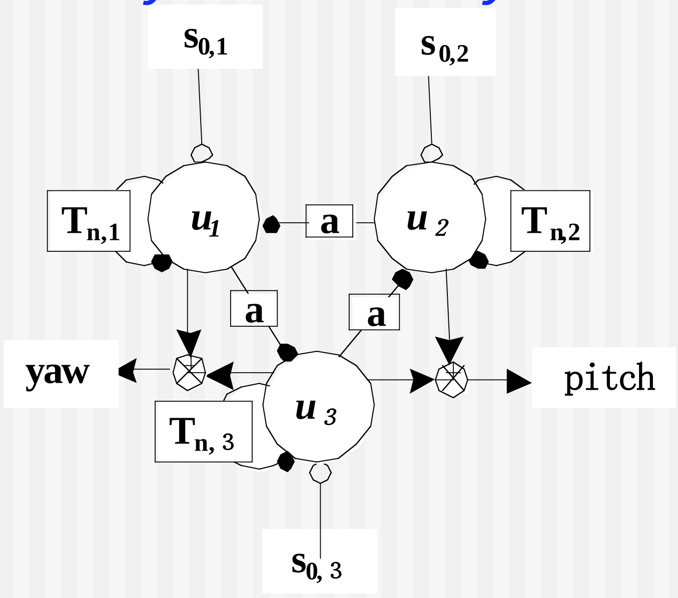
Curvature

Parameters	Value
$u_{0\{e,f\}0}$($u_{0\{e,f\}i}$) (i=1…12)	11.7(9.7)
t	2.0
t'	10.0
b	20.0
w_{fe}	-1.2
w_{ji}	0.2
Cycle [s]	7.0
Phase difference [deg]	60°

(set by trial-and-error)

A New Neural Model (1)
(Cyclic Inhibitory CPG Model)

Unilateral Cyclic Inhibitory CPG Model

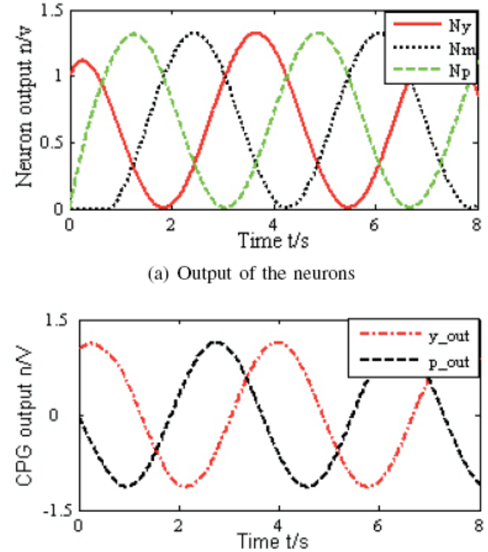


$$T_{n,1}\dot{u}_1 + u_1 = s_{0,1} - ag(u_2)$$
$$T_{n,2}\dot{u}_2 + u_2 = s_{0,2} - ag(u_3)$$
$$T_{n,3}\dot{u}_3 + u_3 = s_{0,3} - ag(u_1)$$
$$y_i = g(u_i), g(u_i) = max(0, u_i) \quad i = 1, 2, 3$$
$$y\_out = y_1 - y_3$$
$$p\_out = y_2 - y_3$$

Theorem 1: Under the condition $T_{n,1}=T_{n,2}=T_{n,3}=\tau$ and $s_{0,1}=s_{0,2}=s_{0,3}=0$, the equations have no stable stationary solution, if and only if

$a \geq 2$ or $a \leq -1$.

Theorem 2: Any solutions of the equations are bounded for t>0 under the condition $a \geq 0$.

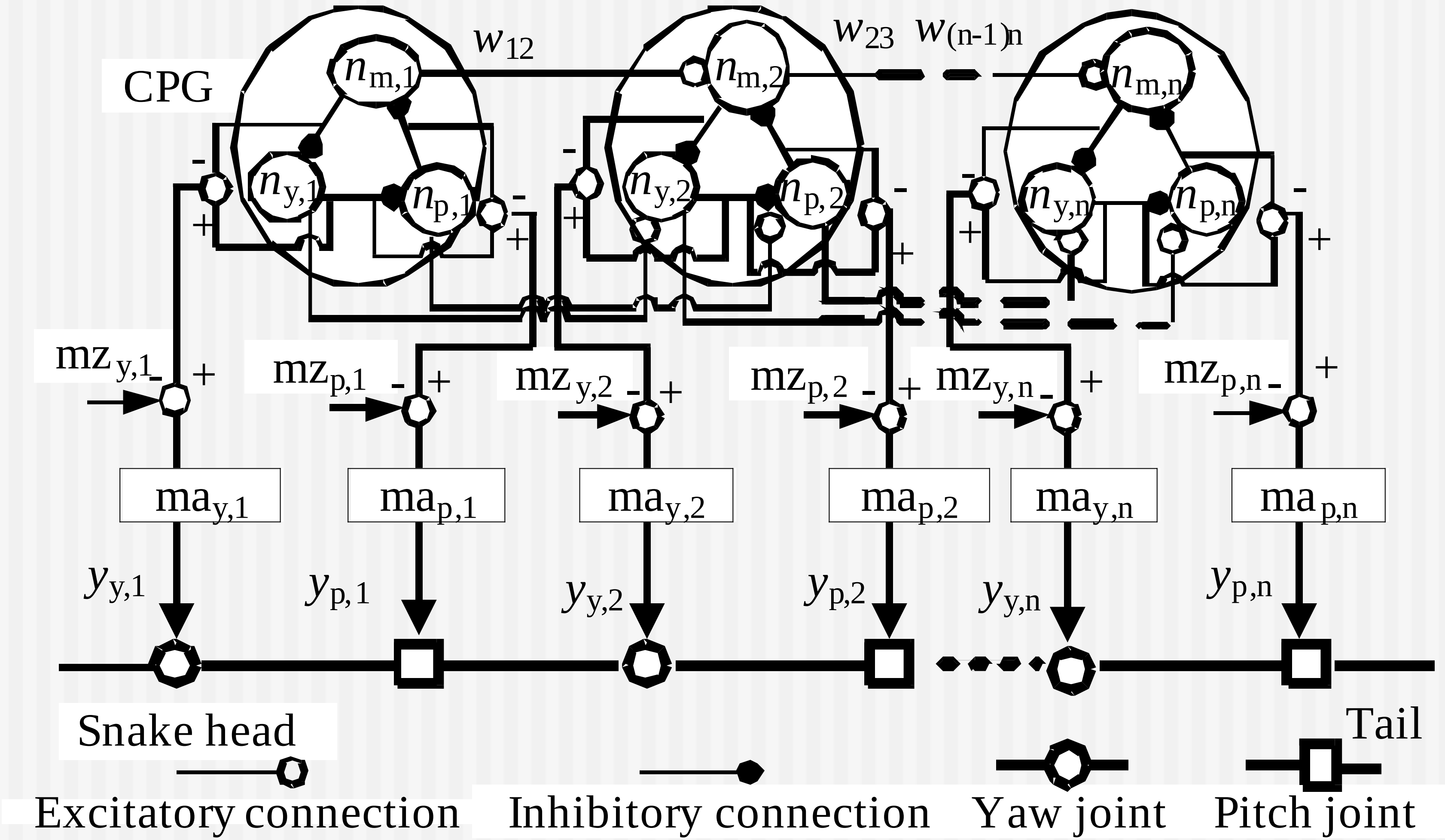


(a) Output of the neurons

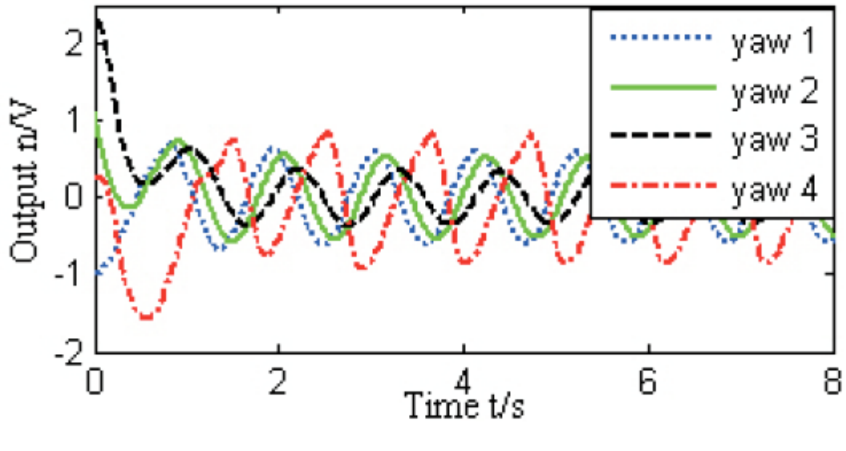
(b) Output of the cyclic inhibitory CPG

Fig. 5. Step response ($T_{n,1}$= $T_{n,2}$= $T_{n,3}$=1, $s_{0,1}$=$s_{0,2}$=$s_{0,3}$=1, a=2)

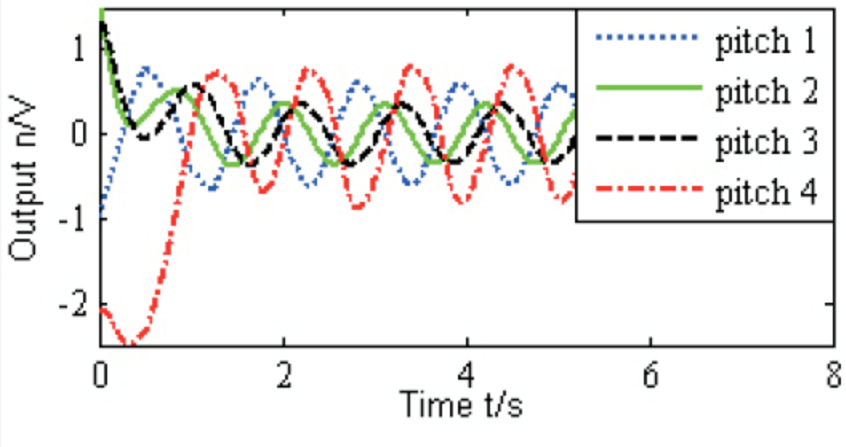
Network Structure and Implementation to Snake Robot



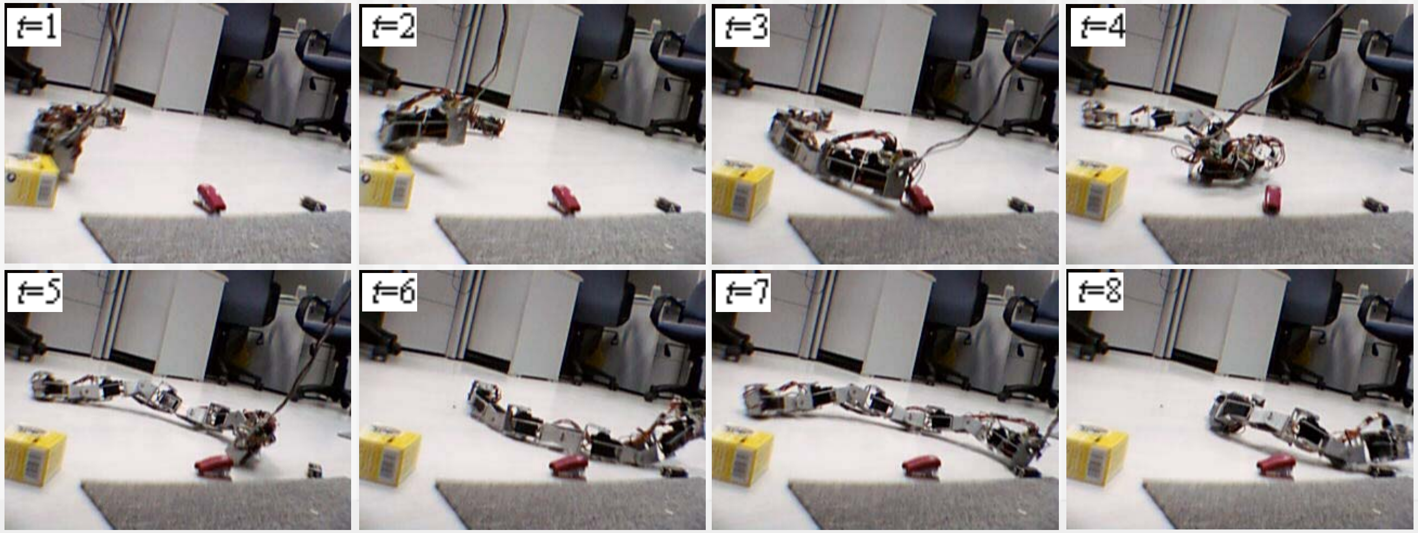
Simulation Result



(a) Yaw joint signal



(b) Pitch joint signal



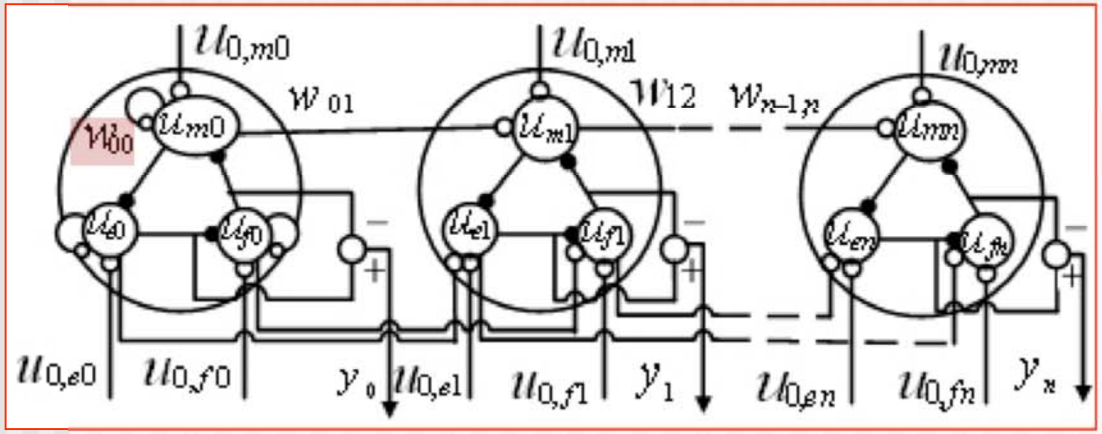
Unilateral Cyclic Inhibitory CPG Model for Serpentine motion of Snake-like Robots

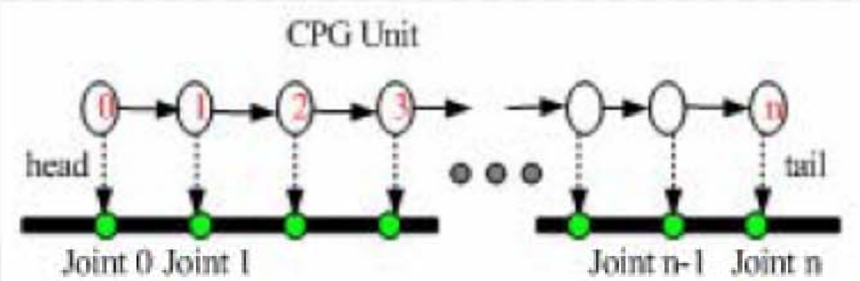
$$\tau\dot{u}\{e,f,m\}_i = -u\{e,f,m\}_i + w_{\{e,f,m\}_i}\, y\{m,e,f\}_i - \beta v\{e,f,m\}_i + u_0\{e,f,m\}_i + Feed\{e,f,m\}_i$$

$$y\{e,f,m\}_i = \max(0, u\{e,f,m\}_i)$$

$$\tau'\dot{v}\{e,f,m\}_i = -v\{e,f,m\}_i + y\{m,e,f\}_i$$

$$y_i = y_{fi} - y_{ei}$$





No pitch, only Yaw
➔ Serpentine Motion

Conditions for a stable oscillation:

$$w_{\{e\}}/(1+\beta) \ge (\mu_0\{m\} - Feed_{\{m\}} - \Sigma\, w_{ij}\, y_j\{m\}) / (\mu_0\{e\} - Feed_{\{e\}} - \Sigma\, w_{ij}\, y_j\{e\}).$$

$$w\{e\} \ge 1 + \tau/\tau'.$$

$$w_{\{m\}}/(1+\beta) \ge (\mu_0\{f\} - Feed_{\{f\}} - \Sigma\, w_{ij}\, y_j\{f\}) / (\mu_0\{m\} - Feed_{\{m\}} - \Sigma\, w_{ij}\, y_j\{m\}).$$

$$w\{m\} \ge 1 + \tau/\tau'.$$

$$w_{\{f\}}/(1+\beta) \ge (\mu_0\{e\} - Feed_{\{e\}} - \Sigma\, w_{ij}\, y_j\{e\}) / (\mu_0\{f\} - Feed_{\{f\}} - \Sigma\, w_{ij}\, y_j\{f\}).$$

$$w\{f\} \ge 1 + \tau/\tau'.$$

$$w_{ij} = \begin{cases} \boxed{w_0} & \text{if} \quad i = j-1 \\ 0 & \text{others} \end{cases}$$

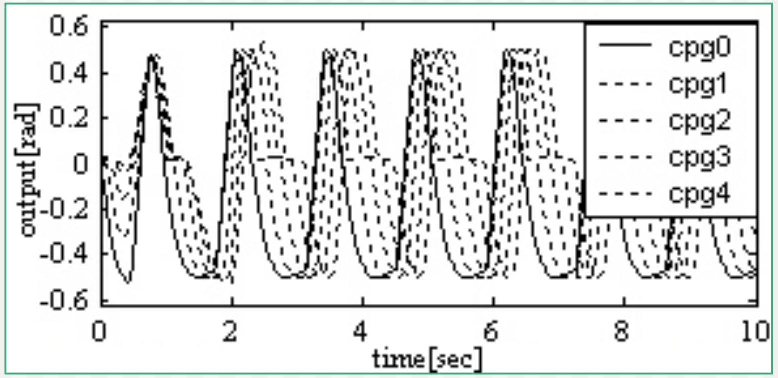
Realization of Serpentine Motion by Unilateral Cyclic Inhibitory CPG Model

Exp.

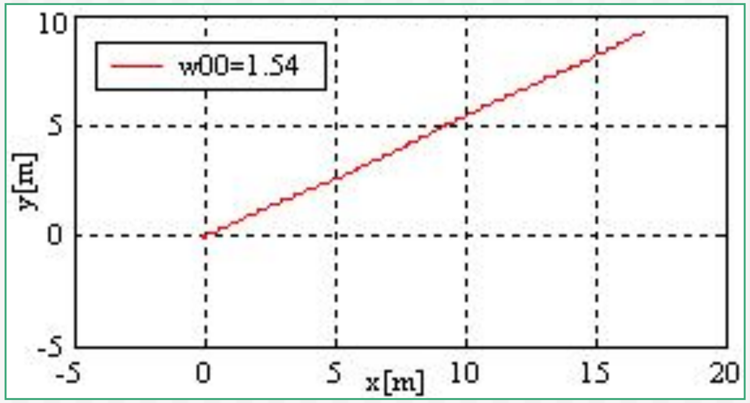
MAIN PARAMETRS OF THE CPG MODEL

Driving input $\mu_0(e,f,m)_i$ (i=0, ... ,n-1)	25
Time constant for state τ	03
Time constant for fatigue τ'	03
Fatigue coefficient β	1
Head-neuron connection weight w_{00}	154
Inter-CPG connection weight w_0	15
Inter-neuron connection weight $w(e,f,m)_i$ (i=1, ... ,n-1)	25

Output curve of CPG



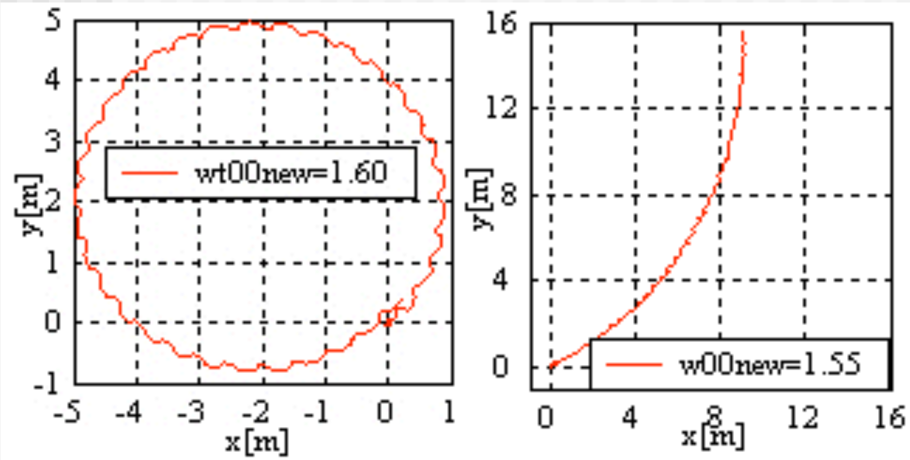
Head trajectory of the robot



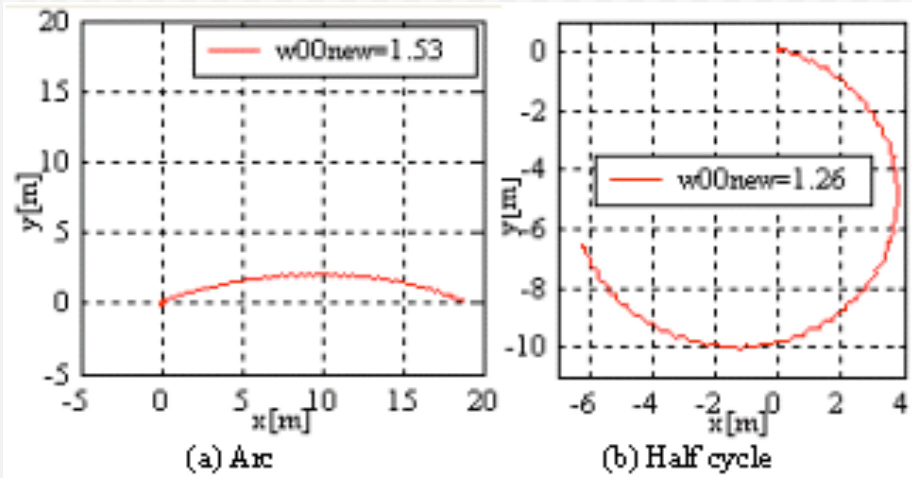
CPG Parameters for Turn Motions

$$w_{00new} = w_{00} + \Delta w_{00}$$

$W_{00}=1.54$



Left-turn motion

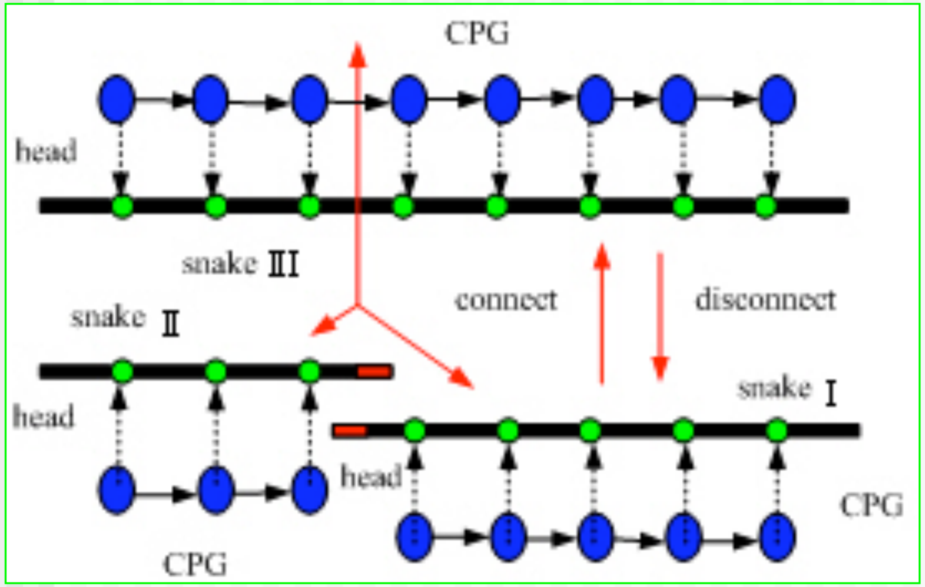


Right-turning motion

1) Δw_{00} >0, turns left (anti-clockwise)
2) Δw_{00} <0, turns right (clockwise)

◆ $|\Delta w_{00}|$ becomes smaller, turn motion angle become smaller

CPG Parameters for Reconfiguration

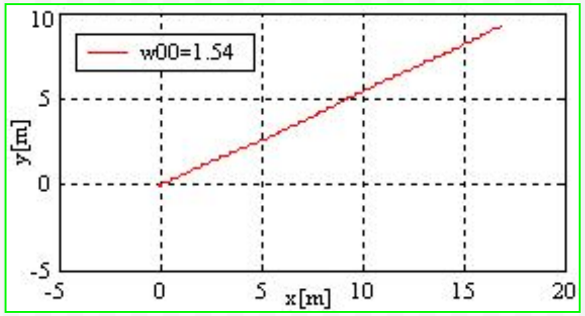


THE PARAMETERS OF THE CPG MODEL

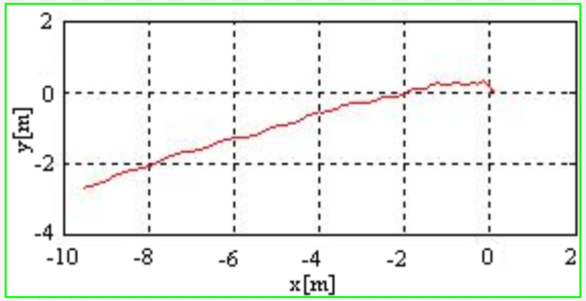
	Model I	Model II	Model III
w_{00}	1.54	0.1	-0.1
w_0	1.5	10	0.3

Head trajectories

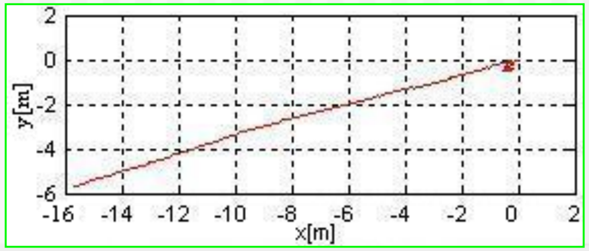
Model I



Model II

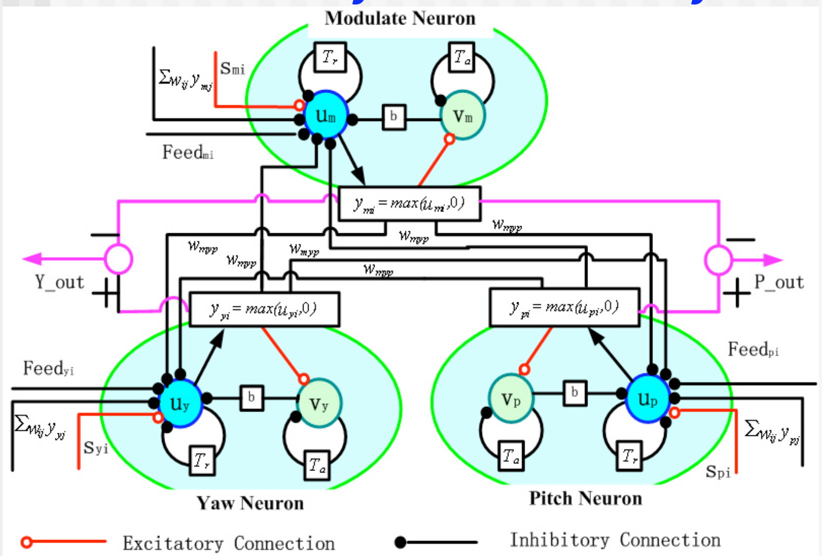


Model III



A New Neural Model (2)
(Cyclic Inhibitory CPG Model)

Bidirectional Cyclic Inhibitory CPG Model



Theorem 1: A solution of equations exists uniquely for any initial state and is bounded for t > 0.

Theorem 2: The equations have at least one stationary solution.

Conditions for a stable oscillation:

$$\frac{w_{ypm}}{1+b} \le min\left(\frac{sin_{\{y,p,m\},i}}{sin_{\{p,m,y\},i}}, \frac{sin_{\{y,p,m\},i}}{sin_{\{m,y,p\},i}}, \frac{sin_{\{p,m,y\},i}}{sin_{\{m,y,p\},i}}, \frac{sin_{\{p,m,y\},i}}{sin_{\{y,p,m\},i}}, \frac{sin_{\{m,y,p\},i}}{sin_{\{y,p,m\},i}}, \frac{sin_{\{m,y,p\},i}}{sin_{\{p,m,y\},i}}\right).$$

$$w_{ypm} > 1 + \frac{T_r}{T_a}.$$

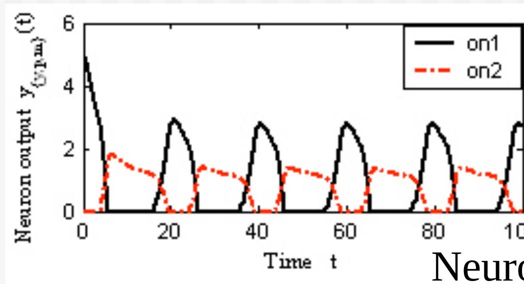
$$T_{n,\{y,p,m\},i}\,\dot{u}_{\{y,p,m\},i} = -u_{\{y,p,m\},i} - w_{\{y,p,m\},i}\,y_{\{p,m,y\},i} - w_{\{y,p,m\},i}\,y_{\{m,y,p\},i} - \beta v_{\{y,p,m\},i} + s_{0,\{y,p,m\},i} + Feed_{\{y,p,m\},i} + \sum_{j=1}^{n} w_{ij}\,y_{\{y,p,m\},j}$$

$$T_{a,\{y,p,m\},i}\,\dot{v}_{\{y,p,m\},i} = -v_{\{y,p,m\},i} + y_{\{y,p,m\},i}$$

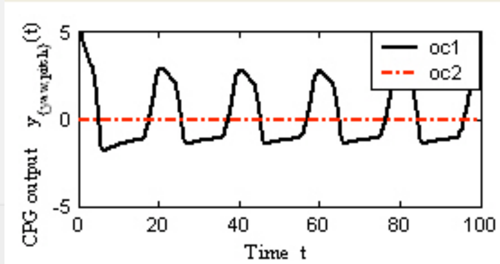
$$y_{\{y,p,m\},i} = g(u_{\{y,p,m\},i}),\quad g(u_{\{y,p,m\},i}) = \max(0,\ u_{\{y,p,m\},i})$$

$$y_{yaw,i} = y_{y,i} - y_{m,i}$$

$$y_{pitch,i} = y_{p,i} - y_{m,i}$$

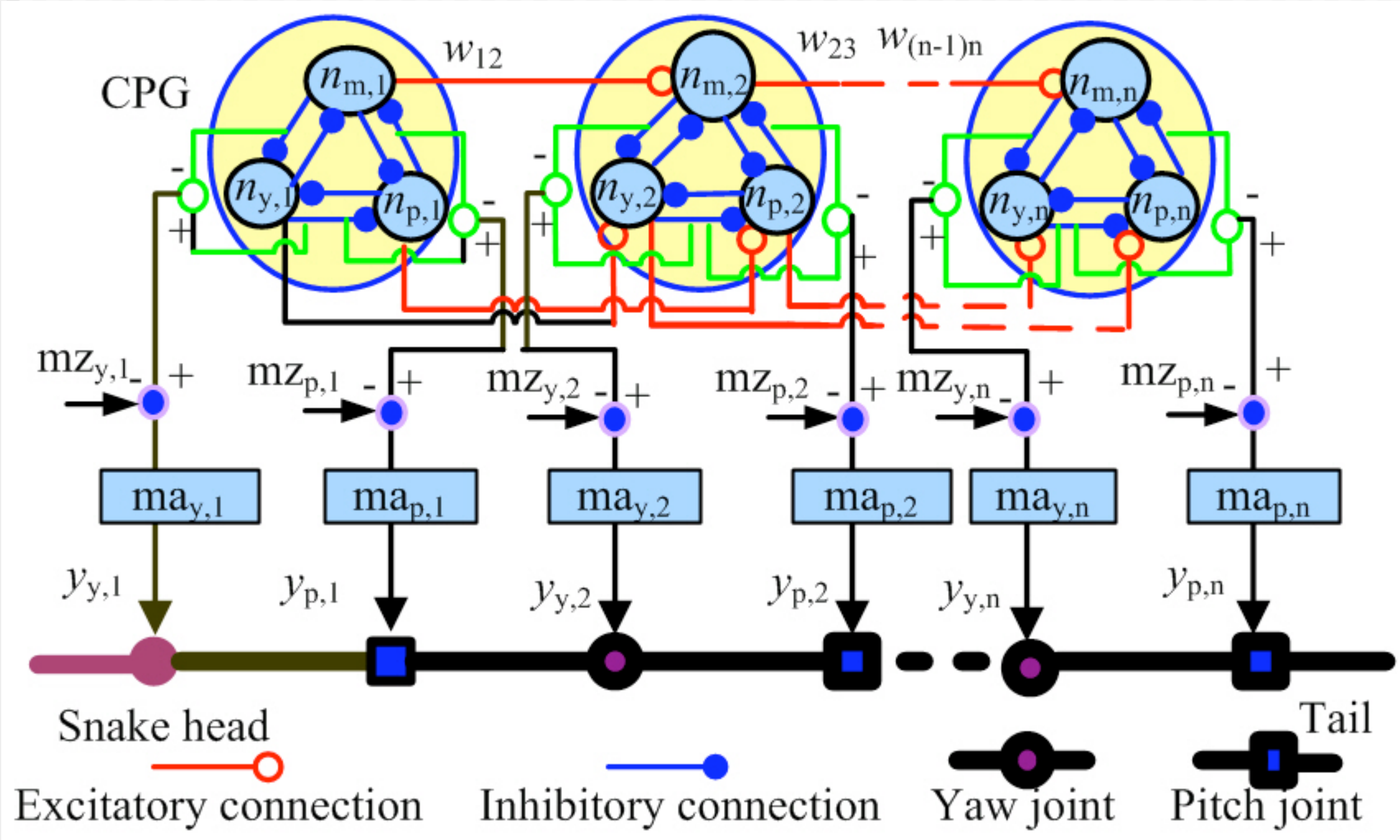


(a)
Neuron ouput in a CPG

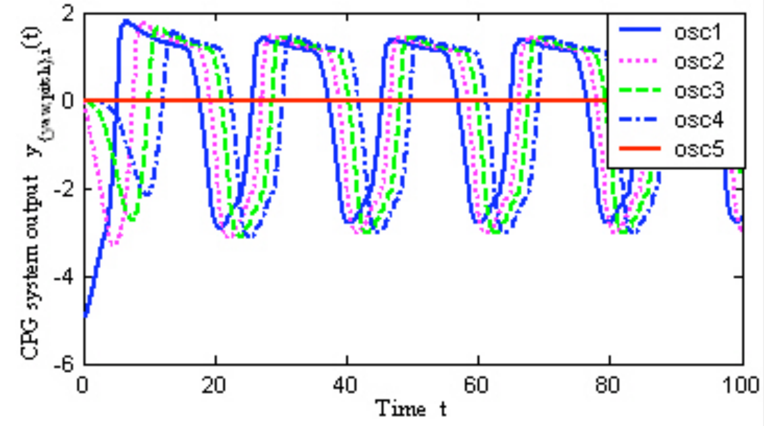


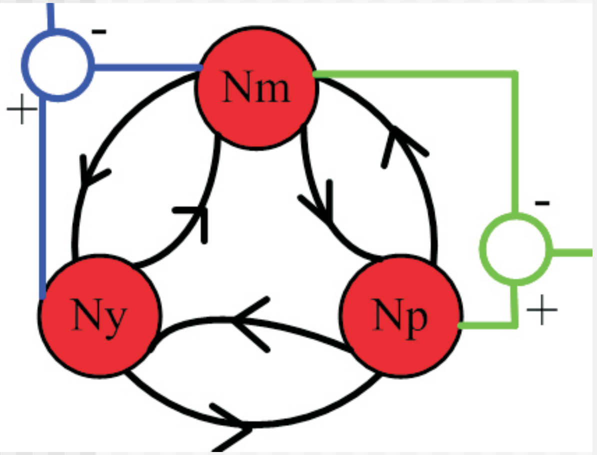
(b)
CPG ouput

Network Structure and Implementation to Snake Robot



Simulation Result





[Y1,P1,M1]

[**1**,0,0]: Serpentine

[0,**1**,0]: Concertina

[0,0,**1**]: Sidewinding