Summary of simulation results

Tony

We consider two degree of freedom manipulator moving in the $xy$ plane

the volume of link $i$ is calculated as $V\_{li}=πR\_{i}^{2}li$

*The added mass* $m\_{ai}$ of a a cylindrical theoretically can be calculated using the following formula

$$m\_{ai}=\rho\pi R\_{i}^{2}l\_{i},\;\;i=1,\,2\\$$

where $ρ=1.01$ is the water density. The total mass of the manipulator’s links are: $m\_{Ti}=m\_{i}+m\_{ai}$.

*Buoyancy* can be calculated by the following equation :

$$B\_{i}=\rho gV\_{li}=b\_{i}g\text{, where: }b\_{i}=\rho V\_{li}\;\;\;i=1,\,2\par
\\$$

*Drag Force* affects each link individually. For simplicity in simulation we assume that water flows along x axis in local coordinate system.



Water velocity

As it is shown in figure ???, at beginning, water velocity is $v\_{w}=[1,0](m/s)$. During the first 30 seconds of the simulation, it will slow down, $v\_{w}^{x}=e^{(−0.0231t)}$. We assume that the robot is exposed to a sudden flow of water with high speed around $2.71(m/s)$. $v\_{w}^{x}=e^{(t−35)}$
The target trajectory : $ϕ\_{d1}=0.14Sin(0.5t),ϕ\_{d2}=0.14Cos(0.5t)$. The simulation is run for 60 seconds.
A neural network of 64 nodes are used for both cases. The manipulator parameters are shown in table ???.

Robot Parameters

**Parameter**

**Explanation**

**Value**

$m\_{1}$

Mass of link 1

2 Kg

$m\_{2}$

Mass of link 2

0.85 Kg

$l\_{1}$

Length of link 1

0.35 m

$l\_{2}$

Length of link 2

0.31 m

$R\_{1}$

Radius of link 1

0.2 m

$R\_{2}$

Radius of link 2

0.2 m

$m\_{a1}$

Added Mass of link 1

$ρπR\_{1}^{2}l\_{1}$

$m\_{a2}$

Added Mass of link 2

$ρπR\_{2}^{2}l\_{2}$

$V\_{1}$

Volume of link 1

$πR\_{1}^{2}l\_{1}$

$V\_{2}$

Volume of link 2

$πR\_{2}^{2}l\_{2}$

# first case

In this case Drag force is approximated with the other robotic dynamics
The Controller is

$$\tau=-e\_{1}-k\_{2}e\_{2}+\hat{W}^{T}N(S)\par
\\$$

The network input is $S=[X\_{1}^{T},X\_{2}^{T},α^{T},\dot{α}^{T}]$, and $δ\_{1}(S)$ is an $n×1$ vector represents the approximation error.
The adaptive law is designed as:

$$\dot{\hat{W}}=-\Gamma\_{i}[N(S)e\_{2,i}+\sigma\_{i}\hat{W}\_{i}],i=1,2,3,...\par
\\$$



Tracking Error of link 1 For the First Case



Tracking Error of link 2 For the First Case



Approximation Error For the First Case



Drag Torque of link 1 and link 2 For the First Case



Input Torques For the First Case

**configuration values**
variance = 50
The center of the radius function is put to one (1)
$σ\_{1}=σ\_{2}=0.02$.
$Γ=20I\_{64×64}$.
The initial weight $^\_{1,i}=^\_{2,i}=0,(i=1,2…,64)$.

# Second case

Drag force is considered as an external disturbance. A nonlinear disturbance observer is used to compensate it. Neural network is used to approximate the other dynamics. The control law:

=-e\_1-K\_2e\_2+^T(S)+\_des

the Neural network is used to approximate the other dynamics as following :

^T(S)= &amp; M(x\_1)+C(x\_1,x\_2)+G(x\_1)

The adaptive law of the neural network proposed in (???) is designed as:

=-\_i[(S)e\_2,i+\_i\_i],i=1,2,3,...

The estimated disturbance is:

$$\frac{\partial\Phi(e\_{2})}{\partial(e\_{2}^{T})}=B\text{ is a }(n\*n)\text{ positive constant matrix.}\nonumber \\$$

$$\hat{f}\_{des}=\hat{e}\_{3}-\Phi(e\_{2})\par
\\$$

and



Tracking Error of link 1 in the second case



Tracking Error of link 2 in the Second Case



Neural Network Approximation Error in the Second Case



Drag Torque and Approximation Error of link 1 in the Second Case



Drag Torque and Approximation Error of link 2 in the Second Case



Input Torques in the Second Case

**Configuration values**
$k\_{1}=diag[5,5],k\_{2}=diag[7,7]$.
$B=diag[20,20]$.
The neural network has 64 nodes.
variance is 100.
The center of the radius function is put to one (1)
In the adaptive low (LABEL:DO\_Adaptive\_Law), $γ\_{1}=γ\_{2}=0.005$
$Ω=100I\_{64×64}$.
The initial weight $^\_{1,i}=^\_{2,i}=0,(i=1,2…,64)$.