Summary of simulation results

Tony

We consider two degree of freedom manipulator moving in the plane

the volume of link is calculated as

*The added mass* 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 is the water density. The total mass of the manipulator’s links are: .

*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 . During the first 30 seconds of the simulation, it will slow down, . We assume that the robot is exposed to a sudden flow of water with high speed around .   
The target trajectory : . 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**

Mass of link 1

2 Kg

Mass of link 2

0.85 Kg

Length of link 1

0.35 m

Length of link 2

0.31 m

Radius of link 1

0.2 m

Radius of link 2

0.2 m

Added Mass of link 1

Added Mass of link 2

Volume of link 1

Volume of link 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 , and is an 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)  
.  
.  
The initial weight .

# 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**  
.  
.  
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),   
.  
The initial weight .