The Robot Linear Dynamic Regressor

The Robot Linear Regressor

- It is always possible to rewrite the dynamics of a robot in the form

$$\boxed{M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \overbrace{\underbrace{Y(q,\dot{q},\ddot{q})}_{N \times p}}^{\text{regression matrix}}\ \overbrace{\underbrace{a}_{p \times 1}}^{a\ =\ \text{vector of dynamic coefficients}} = u}$$

- These parameters are instrumental to control the robot effectively and usually they are not provided by the robot manufacturers
- The **relationship is linear**!

How to Compute the Dynamic Parameters (1)

- At each time step of our simulation we can compute one regressor matrix defined as

$$Y_t(q(t), \dot{q}(t), \ddot{q}(t))a = u(t)$$

- Where Y_t is the regressor matrix at time t, **a** represents the system parameters, and u(t) is the control input at time t

How to Compute the Dynamic Parameters (2)

- Given the regressor matrices Y_t and control inputs u(t) at each time step t from t = 1 to t = T , we can stack these to form:

$$\hat{Y}a = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_T \end{bmatrix} a = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_T \end{bmatrix} = \hat{u}$$

- To solve for a, we compute the pseudoinverse of the stacked regressor matrix

$$a = \hat{Y}^{+}\hat{u}$$

Configuration File

- For this Lab session activate the noise and start with a small noise like in the picture.

```
"noise_flag":[1],
"robot_noise":[
        {
            "joint_cov": 0.0001,
            "joint_vel_cov": 0.0001,
            "joint_acc_cov": 0.0001,
            "base_pos_cov": 0.0001,
            "base_ori_cov": 0.0001,
            "base_lin_vel_cov": 0.0001,
            "base_ang_vel_cov": 0.0001,
            "joint_torque_cov":0.0001
        }
]
```

Laboratory Objectives (1)

- **Parameters Estimation**
  - Write the code to collect the data for computing the dynamic parameters

```
# Compute regressor and store it
cur_regressor = dyn_model.ComputeDynamicRegressor(q_mes, qd_mes, qdd_mes)
```

- **Model Validation**
  - After estimating the parameters, validate the model by comparing the simulated output of the manipulator with the model learned of the estimated parameters
  - Compute the **adjusted R-squared**
  - Compute the **F-Statistics**
  - Compute the **confidence interval** for the parameters and for the prediction

Laboratory Objectives (2)

- **Model Validation with increasing noise**
  - Repeat the first task by increasing the noise on joint_pos, joint_vel and joint_acc by changing the covariance value:
  - **0.01**
  - **0.1**
  - **1**
  - Recompute for each of this value the previous statistics

You can find a script from which you can start working your code in this repo week_12

https://github.com/VModugno/lab_sessions_COMP0245_PUBLIC