Algorithm 3.1 Kernel Disturbance Observer using dynamic regression for parameter estimation

Input: $\mathbf{p_M} = [p_M(t_k - \delta) : p_M(t_k - \delta)]$, $\Theta_\mathbf{M} = [\Theta_M(t_k - \delta) : \Theta_M(t_k)]$

Output: $\hat{d_p}(t_k)$, $\hat{d_\Theta}(t_k)$

1 Set $\mathbf{w}_1^0 = [1, 1, 1, 1]$, $k = 1$

2 Obtain Full State Estimate

3 **for** $\zeta \in [x, y, z, \phi, \theta, \psi]$ **do**

4 **Parameter Estimate for Local Surrogate Model:** Compute $\hat{\mathbf{w}}_{\zeta_k}$

$$\hat{\mathbf{w}}_{\zeta_k} = \underset{\mathbf{w}_{\zeta_k}}{\operatorname{argmin}} \big\{ \gamma ||\mathbf{w}_{\zeta_k} - \hat{\mathbf{w}}_{\zeta_{k-1}}||_2^2 + \lambda ||\mathbf{w}_{\zeta_k}||_2^2 + ||\zeta_M - \zeta_{est}||_2^2 \big\}$$

5 **State Estimation:** Compute state estimate $\hat{\zeta}(t_k)$

$$\hat{\zeta}(t_k) = \frac{1}{((t_k) - (t_k - \delta))^4} \int_{t_k - \delta}^{t_k} K_{DS}(t_k, \tau) \zeta_M(\tau) \mathrm{d}\tau$$

6 **Higher Order Derivatives:** Compute the higher order derivatives $\dot{\hat{\zeta}}(t_k)$ and $\ddot{\hat{\zeta}}(t_k)$ using equations mentioned in Appendix.

7 **end**

8 Compute External Disturbances $\hat{d_p}(t_k) = [\hat{d_x}, \hat{d_y}, \hat{d_z}]$ and $\hat{d_\Theta}(t_k) = [\hat{d_\phi}, \hat{d_\theta}, \hat{d_\psi}]$:

$$\begin{aligned}
\hat{d_\phi} &= \ddot{\hat{\phi}} - (r_1 \dot{\hat{\theta}} \dot{\hat{\psi}} - r_2 \dot{\hat{\theta}} w + q_1 U_2); \quad & \hat{d_x} &= \ddot{\hat{x}} - \Big((C_{\hat{\phi}} S_{\hat{\theta}} C_{\hat{\psi}} + S_{\hat{\phi}} S_{\hat{\psi}}) \frac{1}{m} U_1 \Big) \\
\hat{d_\theta} &= \ddot{\hat{\theta}} - (r_3 \dot{\hat{\phi}} \dot{\hat{\psi}} + r_4 \dot{\hat{\phi}} w + q_2 U_3); \quad & \hat{d_y} &= \ddot{\hat{y}} - \Big((C_{\hat{\phi}} S_{\hat{\theta}} S_{\hat{\psi}} - S_{\hat{\phi}} C_{\hat{\psi}}) \frac{1}{m} U_1 \Big) \\
\hat{d_\psi} &= \ddot{\hat{\psi}} - (r_5 \dot{\hat{\theta}} \dot{\hat{\phi}} + q_3 U_4); \quad & \hat{d_z} &= \ddot{\hat{z}} - \Big(-g + (C_{\hat{\phi}} C_{\hat{\theta}}) \frac{1}{m} U_1 \Big)
\end{aligned}$$

9 Set $k \leftarrow k + 1$ and go to Step 2.

$$d_i = k.a_i(t) sin(\omega t - \phi_i) \tag{1}$$

$$a_1'(t) = 0.15\big(\sigma(a_2 - a_1)\big) \tag{2a}$$

$$a_2'(t) = 0.15\big(\rho a_1 - a_1 a_3 - a_2\big) \tag{2b}$$

$$a_3'(t) = 0.15\big(a_1 a_2 - \beta a_3\big) \tag{2c}$$

$$\hat{\zeta}(t_k) \cong \int_a^b K_{DS}(t_k, \tau)\zeta_M(\tau)d\tau \quad \forall \quad t_k \in [a, b] \tag{3}$$

$$K_{DS}(t_k, \tau) \triangleq \begin{cases} K_{F,\zeta}(t_k, \tau) & for \quad \tau \leq t_k \\ K_{B,\zeta}(t_k, \tau) & for \quad \tau > t_k \end{cases} \tag{4}$$

$$\begin{aligned} &K_{F,\zeta}(t, \tau) \\ &= \frac{1}{(t-a)^4 + (b-t)^4}\Bigg[\Big(16(\tau - a)^3 - a_3(\tau - a)^4\Big) \\ &+ (t-\tau)\Big(- 72(\tau - a)^2 + 12a_3(\tau - a)^3 - a_2(\tau - a)^4\Big) \\ &+ \frac{(t-\tau)^2}{2}\Big(96(\tau - a) - 36a_3(\tau - a)^2 + 8a_2(\tau - a)^3 - a_1(\tau - a)^4\Big) \\ &+ \frac{(t-\tau)^3}{6}\Big(- 24 + 24a_3(\tau - a) - 12a_2(\tau - a)^2 + 4a_1(\tau - a)^3 - a_0(\tau - a)^4\Big)\Bigg] \end{aligned} \tag{5}$$

$$\mathbf{w} := \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^\intercal$$

$\tau - a$ and $t - \tau$ remain constant with respect to the position in the sliding window as time progresses

$$d_\phi \quad d_\theta \quad d_\psi \tag{6}$$

Simulation Time in Window (seconds)

Table 1: Time reductions using vectorization in different languages

Time in milliseconds	Programming Language		
	MATLAB	**Python**	**C++**
Conventional Method	4.082	63.3709	0.02469
Vectorization Method	0.553	0.0601	0.00174
Order of Reduction	7.381	1054.42	14.189

$K_{DS}(\tau)$

$y(t)$ d_ϕ d_θ d_ψ N-m