

Algorithm 3.1 Kernel Disturbance Observer using dynamic regression for parameter estimation

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

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 $\hat{\mathbf{w}}_{\zeta_k}$

$$\hat{\mathbf{w}}_{\zeta_k} = \underset{\mathbf{w}_{\zeta_k}}{\operatorname{argmin}} \{ \gamma \| \mathbf{w}_{\zeta_k} - \hat{\mathbf{w}}_{\zeta_{k-1}} \|_2^2 + \lambda \| \mathbf{w}_{\zeta_k} \|_2^2 + \| \zeta_M - \zeta_{est} \|_2^2 \}$$
- 5 **State Estimation:** Compute state estimate $\hat{\zeta}(t_k)$

$$\hat{\zeta}(t_k) = \frac{1}{((t_k) - (t_k - \delta))^4} \int_{t_k - \delta}^{t_k} K_{DS}(t_k, \tau) \zeta_M(\tau) d\tau$$
- 6 **Higher Order Derivatives:** Compute the higher order derivatives $\hat{\dot{\zeta}}(t_k)$ and $\hat{\ddot{\zeta}}(t_k)$ 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 $\hat{d}_\Theta(t_k) = [\hat{d}_\phi, \hat{d}_\theta, \hat{d}_\psi]$:
$$\begin{aligned} \hat{d}_\phi &= \hat{\dot{\phi}} - (r_1 \hat{\theta} \hat{\psi} - r_2 \hat{\theta} w + q_1 U_2); & \hat{d}_x &= \hat{\dot{x}} - \left((C_{\hat{\phi}} S_{\hat{\theta}} C_{\hat{\psi}} + S_{\hat{\phi}} S_{\hat{\psi}}) \frac{1}{m} U_1 \right) \\ \hat{d}_\theta &= \hat{\dot{\theta}} - (r_3 \hat{\phi} \hat{\psi} + r_4 \hat{\phi} w + q_2 U_3); & \hat{d}_y &= \hat{\dot{y}} - \left((C_{\hat{\phi}} S_{\hat{\theta}} S_{\hat{\psi}} - S_{\hat{\phi}} C_{\hat{\psi}}) \frac{1}{m} U_1 \right) \\ \hat{d}_\psi &= \hat{\dot{\psi}} - (r_5 \hat{\theta} \hat{\phi} + q_3 U_4); & \hat{d}_z &= \hat{\dot{z}} - \left(-g + (C_{\hat{\phi}} C_{\hat{\theta}}) \frac{1}{m} U_1 \right) \end{aligned}$$
- 9 Set $k \leftarrow k + 1$ and go to Step 2.

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

$$a'_1(t) = 0.15(\sigma(a_2 - a_1)) \quad (2a)$$

$$a'_2(t) = 0.15(\rho a_1 - a_1 a_3 - a_2) \quad (2b)$$

$$a'_3(t) = 0.15(a_1 a_2 - \beta a_3) \quad (2c)$$

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

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

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

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

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

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