MANUAL DE ESTUDIO · MECATRÓNICA AUTOMOTRIZ

Compatibilidad electromagnética, plataformas de 800 V y diagnóstico a bordo

Capítulo 14 · En el video: minuto 3:12:39 · Duración 14:29

Objetivo. Identificar cómo se acoplan las perturbaciones, dimensionar un filtro y leer códigos de falla.

Ideas clave

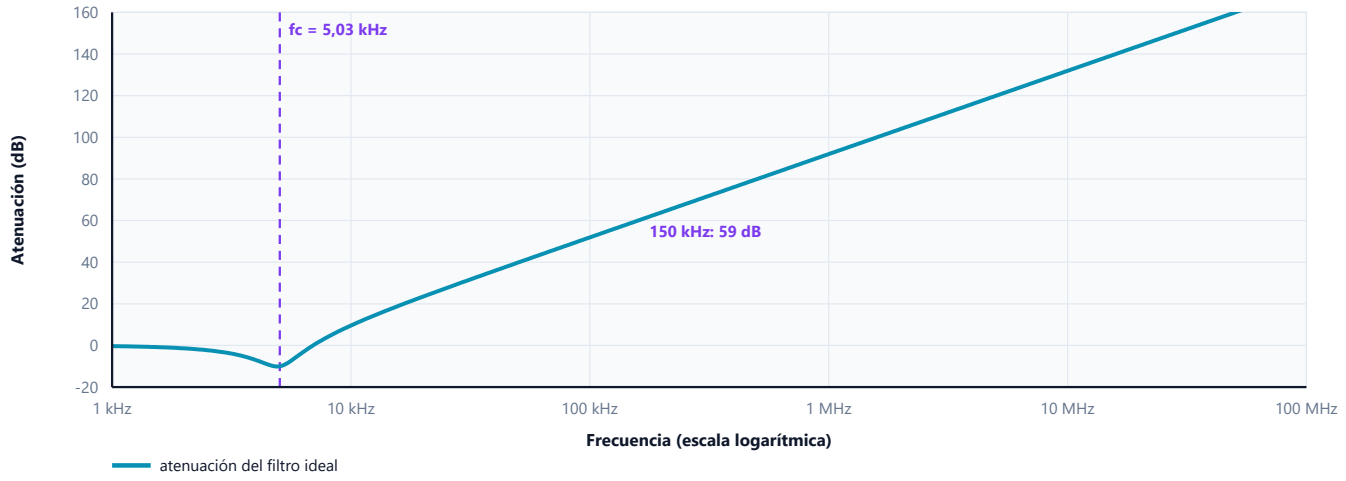
Fuente, acoplamiento y víctima

Toda perturbación tiene una fuente (inversor, encendido, motor), un camino y una víctima (sensor, radio). Los caminos son galvánico (masa compartida), capacitivo (campo eléctrico), inductivo (campo magnético) y radiado.

Transitorios y 800 V

La red de 12 V sufre descargas de carga y arranques en frío. Los inversores de carburo de silicio (SiC) de 400 y 800 V conmutan cientos de voltios en nanosegundos: sus flancos generan ruido de alta frecuencia que se controla con filtros LC, blindajes y buen trazado.

Filtro LC de entrada 10 µH / 100 µF (carga de 1 Ω) cursosmecatronica.com



Cerca de fc el filtro poco amortiguado resuena; sobre ella atenúa 40 dB por década. En la realidad, la inductancia parásita del condensador y la capacidad de la bobina limitan la atenuación sobre unos MHz.

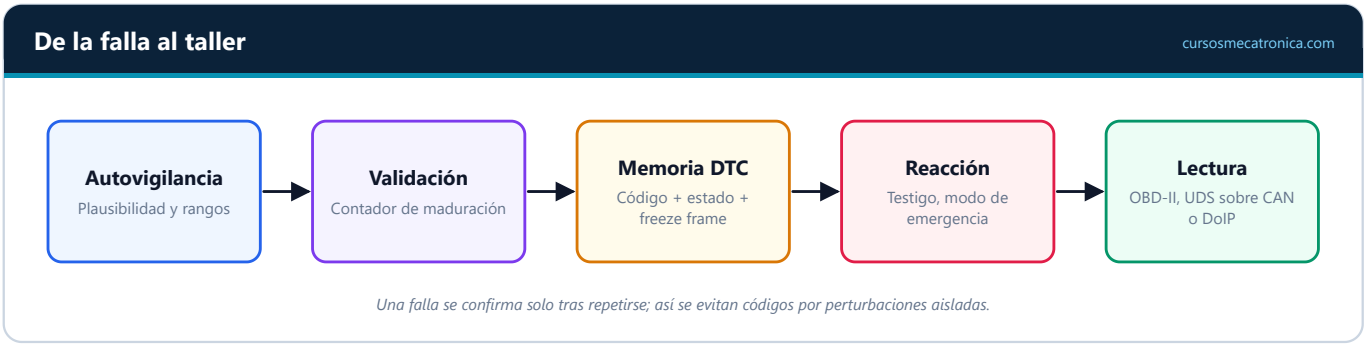
Salida de dtc_y_filtro_cem.py.

Ensayos

Cámara semianecoica para emisiones e inmunidad radiada, inyección de corriente en el cable (BCI), celda TEM; homologación con UNECE R10.

Autovigilancia

La ECU comprueba rangos, plausibilidad entre señales redundantes y sus propias etapas de potencia. Una falla se valida con un contador antes de registrarse con un código (DTC), un byte de estado y una "foto" de las condiciones (freeze frame). Si es grave, el vehículo pasa a modo de emergencia (limp-home).



Diagnóstico a bordo del capítulo 14.

OBD y UDS

OBD-II y EOBD vigilan todo lo que afecta a las emisiones y encienden el testigo; Euro 7 agrega vigilancia continua (OBM). En taller se usa UDS (ISO 14229) sobre CAN o sobre Ethernet (DoIP): lectura de códigos, valores, rutinas, programación de ECU y acceso con seguridad.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Frecuencia de corte LC	$f_c = 1 / (2\pi\sqrt{(LC)})$	10 µH y 100 µF: 5,03 kHz.
Atenuación ideal	$A \approx 40 \cdot \log_{10}(f/f_c)$ dB	40 dB por década.
Acoplamiento capacitivo	$i = C \cdot dv/dt$	Flancos rápidos de SiC.
DTC OBD	2 bytes: letra (P, C, B, U) + 4 dígitos	P0301 = falla de encendido cil. 1.

Ejemplo resuelto: filtro de entrada y lectura de códigos

1. L = 10 µH, C = 100 µF: $f_c = 1/(2\pi\sqrt{(10^{-9})}) = 5{,}03$ kHz.
2. A 150 kHz (inicio de la banda de radio de onda larga en ensayos) atenúa 59 dB; a 1 MHz, 92 dB (ideal).
3. Respuesta UDS 59 02 FF 03 01 00 2F: código P0301-00 con estado 0x2F.
4. Bits de 0x2F: falla ahora, falla en este ciclo, pendiente, confirmada y fallada desde el último borrado.

El filtro ideal no basta sobre unos MHz por los elementos parásitos. dtc_y_filtro_cem.py decodifica también un código de red U0123 pendiente.

Byte de estado de un DTC UDS

Bit	Nombre	Significado
0	testFailed	Falla presente ahora
1	testFailedThisOperationCycle	Falló en este ciclo
2	pendingDTC	Pendiente de confirmar
3	confirmedDTC	Confirmada y memorizada
4	testNotCompletedSinceLastClear	Sin probar desde el borrado
5	testFailedSinceLastClear	Falló desde el borrado
6	testNotCompletedThisOperationCycle	Sin probar en este ciclo
7	warningIndicatorRequested	Pide encender el testigo

Código del capítulo

Los archivos completos vienen en el paquete ZIP del capítulo (carpeta `codigo/`).

dtc_y_filtro_cem.py — atenuación de un filtro LC y decodificación de códigos OBD y UDS con su estado.

```
"""
dtc_y_filtro_cem.py — Capítulo 14: atenuación de un filtro LC de entrada y decodificación
de códigos de falla OBD y UDS.
"""
import math


def atenuacion_db(f, l_h, c_f, r_carga=1.0):
    w = 2 * math.pi * f
    real = 1 - w * w * l_h * c_f
    imag = w * l_h / r_carga
    return -20 * math.log10(1 / math.hypot(real, imag))


def obd_dtc(byte_a, byte_b):
    letra = 'PCBU'[byte_a >> 6]
    return f'{letra}{(byte_a >> 4) & 3}{byte_a & 0xF:X}{byte_b >> 4:X}{byte_b & 0xF:X}'


ESTADO = ['testFailed', 'testFailedThisOperationCycle', 'pendingDTC', 'confirmedDTC',
          'testNotCompletedSinceLastClear', 'testFailedSinceLastClear',
          'testNotCompletedThisOperationCycle', 'warningIndicatorRequested']


def uds_19_02(respuesta):
    """Respuesta positiva a ReadDTCInformation (0x19 0x02): 59 02 máscara [DTC(3) estado(1)]..."""
    assert respuesta[0] == 0x59 and respuesta[1] == 0x02
    salida = []
    for k in range(3, len(respuesta), 4):
        hi, mid, low, st = respuesta[k:k + 4]
        codigo = obd_dtc(hi, mid) + f'-{low:02X}'
        bits = [ESTADO[b] for b in range(8) if st >> b & 1]
        salida.append((codigo, st, bits))
    return salida


if __name__ == '__main__':
    l, c = 10e-6, 100e-6
```

```
    fc = 1 / (2 * math.pi * math.sqrt(l * c))
    print(f'Filtro LC 10 µH / 100 µF: frecuencia de corte {fc / 1000:.2f} kHz')
    for f in (150e3, 1e6, 30e6):
        print(f'  {f / 1e6:6.2f} MHz: atenuación {atenuacion_db(f, l, c):5.1f} dB'
              f' (ideal −40 dB/década: {40 * math.log10(f / fc):5.1f} dB)')

    print(f'\nOBD modo 03, bytes 03 01 -> {obd_dtc(0x03, 0x01)}  (falla de encendido cilindro 1)')
    print(f'OBD modo 03, bytes 41 23 -> {obd_dtc(0x41, 0x23)}')
    resp = [0x59, 0x02, 0xFF, 0x03, 0x01, 0x00, 0x2F, 0xC1, 0x23, 0x87, 0x24]
    for codigo, st, bits in uds_19_02(resp):
        print(f'UDS {codigo} estado 0x{st:02X}: {", ".join(bits)}')
```

Errores frecuentes

- Borrar códigos sin anotar el freeze frame.
- Unir masas de potencia y de señal en cualquier punto (acoplamiento galvánico).
- Calcular el filtro con componentes ideales y olvidar sus parásitos.

Autoevaluación

1. ¿Cuáles son los cuatro caminos de acoplamiento?
2. ¿Qué norma define UDS?
3. ¿Qué indica el bit 3 del estado?

Respuestas. 1. Galvánico, capacitivo, inductivo y radiado. **2.** ISO 14229. **3.** Falla confirmada.