graph_mm

May 25, 2025

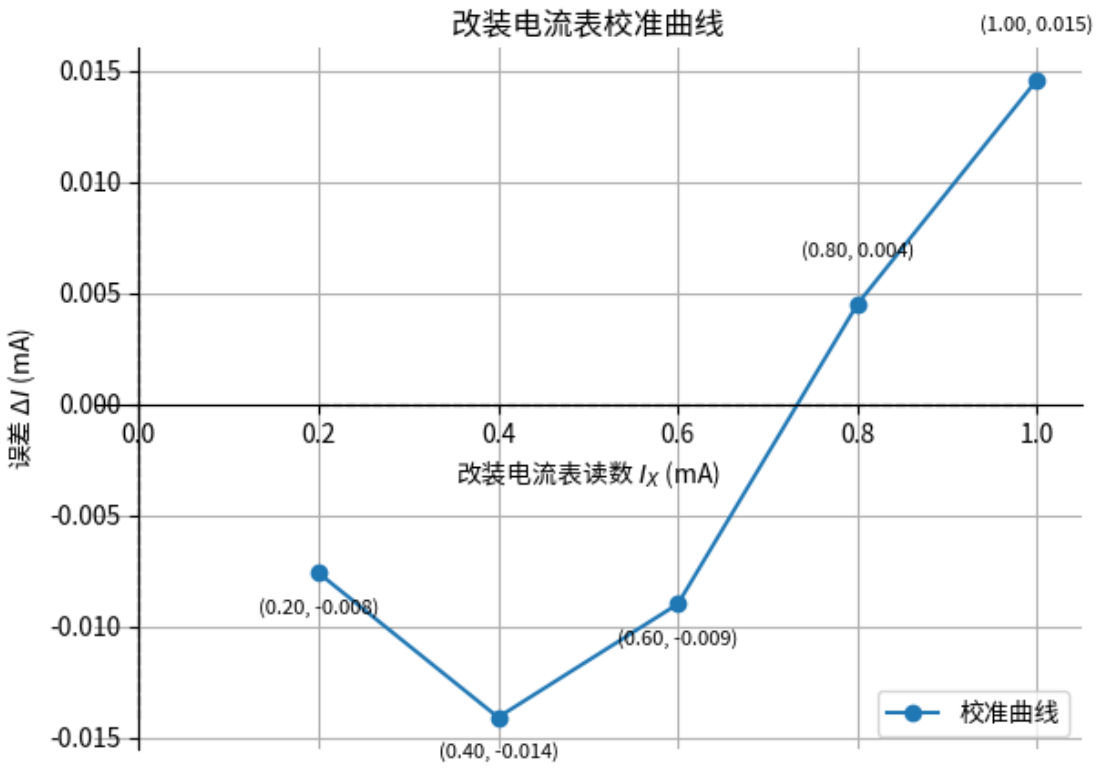
1

- I_X (mA)
- ΔI (mA)

processed_data/ .csv

[19]:
```
import pandas as pd
import matplotlib.pyplot as plt

#
plt.rcParams['font.sans-serif'] = ['Arial Unicode MS', 'Noto Sans SC',␣
 ↪'STSong', 'SimHei', 'sans-serif']
plt.rcParams['axes.unicode_minus'] = False

#
file = 'processed_data/ .csv'
df = pd.read_csv(file)

#
ix = pd.to_numeric(df[' I_X/mA'], errors='coerce')
delta_i = pd.to_numeric(df[' ΔI/mA (ΔI = I_0 - I_X)'], errors='coerce')

fig, ax = plt.subplots(figsize=(7,5))
ax.plot(ix, delta_i, 'o-', label=' ')
#
x_min, x_max = ix.min(), ix.max()
y_min, y_max = delta_i.min(), delta_i.max()
ax.plot([x_min, x_max], [0, 0], color='gray', linestyle='--', lw=1, zorder=1)
ax.plot([0, 0], [y_min, y_max], color='gray', linestyle='--', lw=1, zorder=1)

#
for x, y in zip(ix, delta_i):
    offset = 0.002 if y >= 0 else -0.002
    ax.text(x, y + offset, f'({x:.2f}, {y:.3f})', fontsize=8, ha='center',␣
 ↪va='bottom')

ax.set_xlabel(r' $I_X$ (mA)')
```

```
ax.set_ylabel(r'  $\Delta I$ (mA)')
ax.set_title('     ')
ax.legend()
ax.grid(True)
#
ax.spines['left'].set_position(('data', 0))
ax.spines['bottom'].set_position(('data', 0))
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
plt.show()
```



[20]:
```
#
max_error = delta_i.abs().max()
print(f'   |ΔI| = {max_error:.4f} mA')

#
# 0.2      ±0.2%
# 0.5      ±0.5%
# 1.0      ±1.0%

full_scale = ix.max()  #
if max_error <= full_scale * 0.002:
```

```
    level = '0.2 '
elif max_error <= full_scale * 0.005:
    level = '0.5 '
elif max_error <= full_scale * 0.01:
    level = '1.0 '
else:
    level = ' 1.0 '
print(f'   {level}')
```

```
|ΔI| = 0.0146 mA
 1.0
```

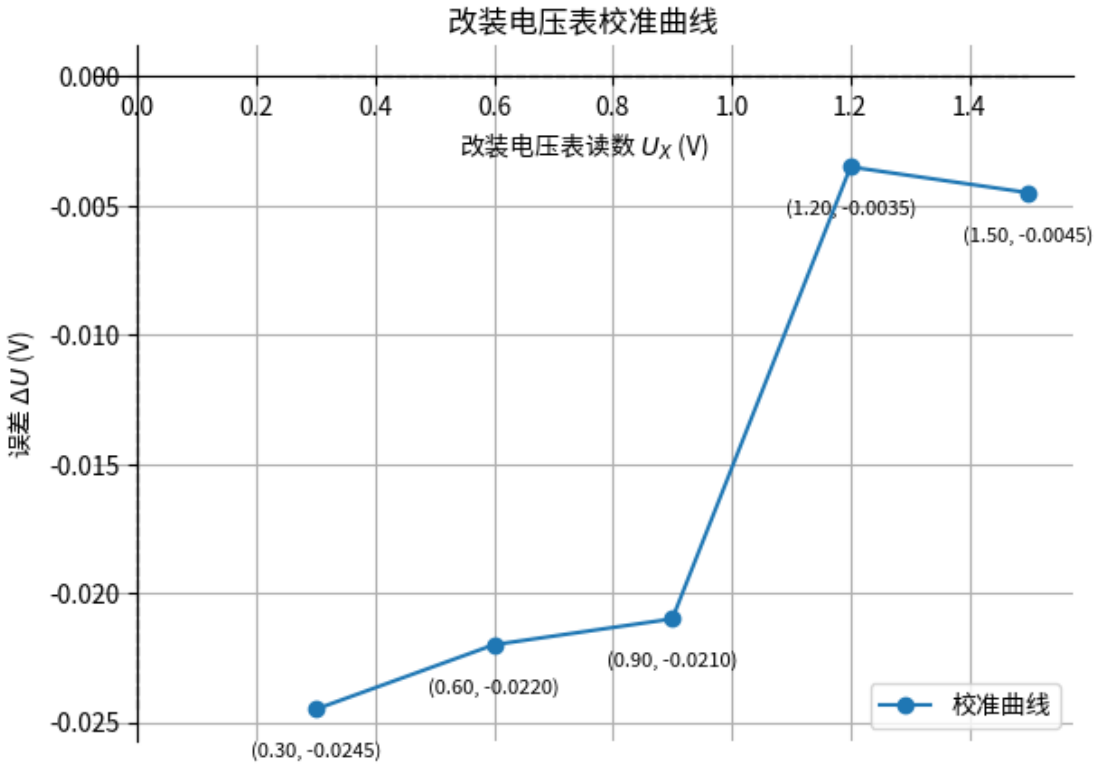
2

- U_X (V)
- ΔU (V)

processed_data/ .csv

[21]:
```
import pandas as pd
import matplotlib.pyplot as plt

#
plt.rcParams['font.sans-serif'] = ['Arial Unicode MS', 'Noto Sans SC',␣
 ↪'STSong', 'SimHei', 'sans-serif']
plt.rcParams['axes.unicode_minus'] = False

#
file_v = 'processed_data/    .csv'
df_v = pd.read_csv(file_v)

#
ux = pd.to_numeric(df_v['    U_X/V'], errors='coerce')
delta_u = pd.to_numeric(df_v['  ΔU/V (ΔU = U_0 - U_X)'], errors='coerce')

fig, ax = plt.subplots(figsize=(7,5))
ax.plot(ux, delta_u, 'o-', label='  ')
#
x_min, x_max = ux.min(), ux.max()
y_min, y_max = delta_u.min(), delta_u.max()
ax.plot([x_min, x_max], [0, 0], color='gray', linestyle='--', lw=1, zorder=1)
ax.plot([0, 0], [y_min, y_max], color='gray', linestyle='--', lw=1, zorder=1)

#
for x, y in zip(ux, delta_u):
    offset = 0.002 if y >= 0 else -0.002
    ax.text(x, y + offset, f'({x:.2f}, {y:.4f})', fontsize=8, ha='center',␣
 ↪va='bottom')
```

```
ax.set_xlabel(r'     $U_X$ (V)')
ax.set_ylabel(r'  $\Delta U$ (V)')
ax.set_title('     ')
ax.legend(loc='lower right')
ax.grid(True)
#
ax.spines['left'].set_position(('data', 0))
ax.spines['bottom'].set_position(('data', 0))
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
plt.show()
```



[22]:
```
#
max_error_u = delta_u.abs().max()
print(f'    |ΔU| = {max_error_u:.4f} V')

#
# 0.2       ±0.2%
# 0.5       ±0.5%
# 1.0       ±1.0%
```

```
full_scale_u = ux.max()  #
if max_error_u <= full_scale_u * 0.002:
    level_u = '0.2 '
elif max_error_u <= full_scale_u * 0.005:
    level_u = '0.5 '
elif max_error_u <= full_scale_u * 0.01:
    level_u = '1.0 '
else:
    level_u = ' 1.0 '
print(f'   {level_u}')
```

```
   |ΔU| = 0.0245 V
    1.0
```