



椿一生 11:09

@ZIKUUBot CNCを作る時に、クローズドループステッパーマーターを使った場合とサーボモーターを使った場合での、性能の差を教えて。



@椿一生 @ZIKUUBot CNCを作る時に、クローズドループステ...

ZIKUUBot **アプリ** 11:09
クローズドループステッパーマーター (CL-stepper) とサーボモーター (servo) を CNC で使用したときの主な性能差は、**精度・応答性・制御方式・負荷耐性・コスト**の5点に集約されます。

|項目|ステッパ|サーボ|

|-----|-----|-----|

|**制御方式**|オープンループ (多くは) +可視化された閉ループ補正|位置・速度・トルクをフィードバックで正確に制御 (PID など) |

|**精度 (分解)**|位置分解はステップ数×ピッチで決まり、ステップ欠損が生じる場合がある|位置誤差をフィードバックで補正でき、欠損無しで高精度|

|**応答性**|高トルクを得るには低速でステップを減らないと不安定|高速・高トルクでも制御が安定|

|**負荷耐性**|大負荷でトルクが減少、オーバースピードで振動|大負荷・高速でもトルク保持|

|**コスト・実装**|パルス制御で構成がシンプル、機械的摩耗も少ない|センサー、アクチュエータ、制御回路が複雑でコスト上乘|

まとめ

- **ステッパ**は構造がシンプルでコストが安いですが、速度上限・負荷上限があり、ステップ欠損が発生すると精度が落ちる。
- **サーボ**はフィードバックにより正確な制御ができるので、速度・負荷どちらも高く、欠損が生じにくく、精度が安定だが、制御回路とコストが高い。

したがって、CNC で高速度・高精度・高負荷が求められる場合はサーボ、低コストで大まわらない精度が許容される場合はステッパが選ばれます。

LLM Broker

LLM_PROFILE LLM_BROKER_URL docker-compose

```
async def get_llm_response(messages_to_send):  
    """Gets a response from LLM Broker."""  
    payload = {  
        "profile": LLM_PROFILE,  
        "messages": messages_to_send,  
    }  
  
    try:  
        async with httpx.AsyncClient(timeout=TIMEOUT) as client:  
            response = await asyncio.wait_for(  
                client.post(f"{LLM_BROKER_URL}/v1/chat", json=payload),  
                timeout=TIMEOUT,  
            )  
  
            response.raise_for_status()  
            data = response.json()  
  
            meta = data.get("meta") or data.get("metadata") or {}  
            if isinstance(meta, dict):  
                logger.info(  
                    "llm_call_done: profile=%s backend=%s model=%s",  
                    data.get("profile") or meta.get("profile") or LLM_PROFILE,  
                    data.get("backend") or meta.get("backend"),  
                    data.get("model") or meta.get("model"),  
                )  
            else:  
                logger.info("llm_call_done: profile=%s", LLM_PROFILE)  
  
            return extract_broker_text(data).strip()  
  
        except asyncio.TimeoutError:  
            logger.warning("llm_call_timeout: profile=%s timeout=%.1f", LLM_PROFILE,  
                return "HTTP {e.response.status_code}"  
  
        except httpx.HTTPStatusError as e:  
            body = e.response.text[:500] if e.response is not None else ""  
            logger.error(  
                "llm_call_http_error: status=%s profile=%s body=%r",  
                e.response.status_code if e.response is not None else None,  
                LLM_PROFILE,  
                body,  
            )  
            return f"LLM Broker HTTP {e.response.status_code}"  
  
        except Exception as e:  
            logger.exception("llm_call_error: profile=%s error=%s", LLM_PROFILE, e)  
            return f"LLM Broker Exception: {e}"
```

LLM_PROFILE`•LLM Broker`•Şç®ıç•?ã•?ã•!ã•?ã??ã?•ã•?ã•®ã??ã?;ã?ãã?«ã•«è"ã®?ã•?ã??ã•!ã•
?ã??ã??ã•®ã•Şã?•Alã?•ã?£ã??ã??ã??ã??ã??ã??ã?ã?½¿ã•£ã•!ã?ã??ã•®ã?ã?•text_conversationã"ã•?ã•
?ã??ã?ã??ã?£ã?¼ã?«ã•Şã•?ã??

```
# llm-broker/backend/profiles.yaml  
profiles:
```

```
  text_extract:  
    backend: llamacpp  
    base_url: http://xxxxxx  
    model: gpt-oss-20b-Q4_K_M  
    temperature: 0.1  
    timeout_sec: 360
```

```
  text_conversation:  
    backend: llamacpp  
    base_url: http://xxxxxx  
    model: gpt-oss-20b-Q4_K_M  
    temperature: 0.7  
    timeout_sec: 360
```

```
  text_extract_ollama_fallback:  
    backend: ollama  
    base_url: http://xxxxxx  
    model: gpt-oss:20b  
    temperature: 0.1  
    timeout_sec: 360
```

```
  text_conversation_ollama_fallback:  
    backend: ollama  
    base_url: http://xxxxxx  
    model: gpt-oss:20b  
    temperature: 0.7  
    timeout_sec: 360
```

Category

- æ?¥ã? ã•®æ'»ã??

Tags

- AI
- Discord
- LLM
- LLM Broker
- ã,?é??ã??ã,?
- ã±±æç"ç??

Date Created

2026ã'7æ??7æ?¥

Author

kazuo-tsubaki