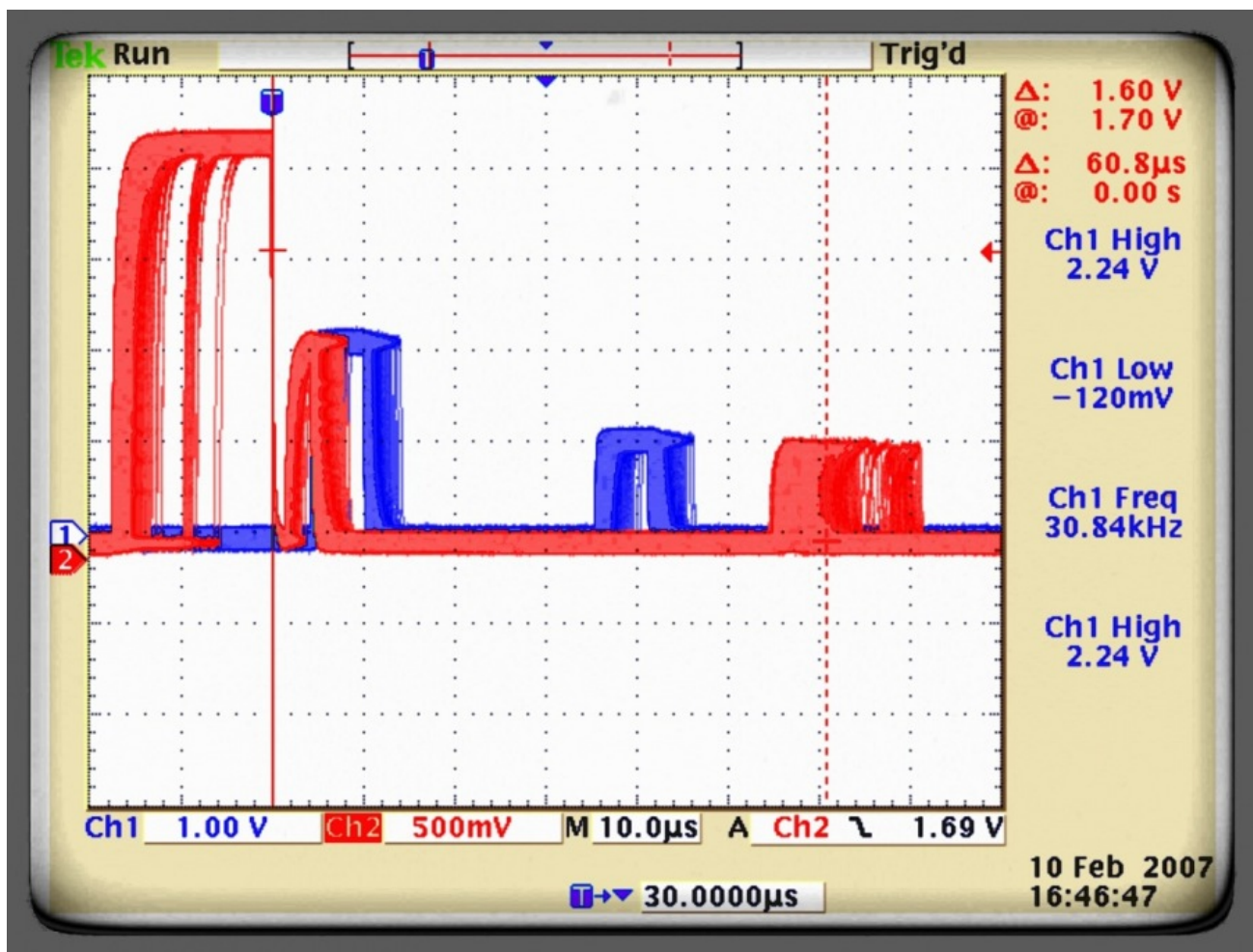


L'importanza dello jitter nei sistemi real-time

bebee.com/producer/l-importanza-dello-jitter-nei-sistemi-real-time



Published on December 17, 2017 on LinkedIn

Introduzione

Interessante la lettura dell'articolo segnalato di Elettronica Open Source intitolato

- [Il problema dei sovraccarichi nei sistemi real-time.](#)

L'importanza dello jitter

Molti sistemi real-time si occupano di controllare la latenza che nel contesto dell'articolo qui sotto riportato è l'analogo dell'esecuzione di un task in ritardo rispetto all'istante pianificato e/o inizio per cui concluda oltre alla dead-line accettabile.

Non è solo questione di buone pratiche di programmazione, ad. es. uso dei semafori [link] oppure di utilizzare un sofisticato sistema di CPU/task scheduling.

Ad esempio, per rendere KISS (keep it as simple as stupid) lo scheduler, quindi renderlo più prevedibile per non dire assolutamente prevedibile ed evitare casi limite e/o imprevisti, è necessario concentrare la nostra attenzione sul concetto di jitter ovvero sulla variabilità della latenza.

Perché più ancora di sapere quanto un task impieghi è importante conoscere quale sia la sua variabilità nell'esecuzione non interrotta.

Estendendo il concetto, esso ci porta anche a conoscere la variabilità delle interruzioni e quindi a sviluppare un modello dinamico completo.

Conoscere lo jitter aiuta ma non ci risolve tout-court il problema delle priorità e della loro inversione. Però, in taluni casi chiusi (not general purpose but dedicated) ci permette di definire un set statico di regole che possano garantire i servizi real-time.

Uno di questi casi è descritto nella mia tesi

- [Ethernet Real-Time per Controllo Motori](#)

di cui sono disponibili anche i codici sorgenti a questo link

- roberto.foglietta.name/mfosset

Comunicare é già controllare

Se ne conviene che, una volta approntato un sistema di comunicazione che rispetti stringenti requisiti di tempistica hard real-time, é possibile costruire su di esso anche un sistema di controllo che invece di sfruttare l'alta capacità di calcolo centralizzata di una CPU sia orientato alla distribuzione dell'intelligenza.

Questo non significa utilizzare diverse CPU meno potenti che insieme alle loro board creerebbero oltre che un innalzamento dei costi anche un maggiore rischio di fallimento hardware, perché i anche l'hardware si rompe, ma a una più equilibrata distribuzione dell'intelligenza nei controlli e nei sensori tramite l'uso di PIC.

Ad esempio, rispetto all'articolo citato e la lettura mancata dell'encoder avremmo un PIC sul motore che si occupa del solo task del controllo movimento: parametri di movimento in input, stato del movimento in output.

Questo implica che il sistema di comunicazione in tempo reale abbia anche un elemento intrinseco di misura dei tempo ovvero analogo a una sincronizzazione fine degli orologi.

Perciò lo jitter diventa un elemento essenziale del controllo in quanto i quarzi derivano, sia come frequenza, sia come RTC orologio interno, ma nonostante questo i cicli di comunicazione sincrona possono essere rigorosamente rispettati.

Perché le derive termiche dei quarzi, anche se non fossero compensate on-the-board, sono lente rispetto ai tempi di reazione del ciclo principale di controllo: secondi vs microsecondi.

Perciò il controllo principale deve gestire una lista circolare di attività (token ring task chain) che nel suo complesso determina la business logic del sistema.

Utilizzando questo approccio e lasciando liberi alcuni slot per le azioni non ricorrenti avremo implementato una logica di funzionamento statica quindi con risorse di calcolo e tempo pre-allocate ma rimanendo aperti alla possibilità di gestire variazioni e/o eccezioni.

Le variazioni dinamiche della business logic saranno implementate come una rivisitazione dell'insieme di azioni di ciclo in ciclo.

Invece, le eccezioni estemporanee e immediate saranno gestite allocandole negli slot lasciati liberi per queste evenienze.

Non ci sono infatti dubbi che non si possa pretendere la garanzia hard real time e nel contempo sfruttare la potenza di calcolo al 100%.

L'allocazione delle risorse dovrebbe attestarsi sull'80% degli slot/time per il ciclo di riferimento e al 96% per il ciclo di massimo impegno entro 3 sigma.

Le percentuali possono variare, le aspettative sulla confidenza del controllo posso variare, da caso a caso, ma la struttura concettuale di base rimane.

Confidenza del controllo

Quando si parla di progettazione di sistemi di controllo gli aspetti rilevanti si trovano sugli indici di secondo ordine, jitter e confidenza, perché quelli di primo ordine sono già dipendenti dalla logica di funzionamento mentre quelli fondamentali sono dettati dai vincoli di sistema.

Perciò il controllo di un sistema di controllo non può che essere fatto sugli indicatori di secondo livello. Il sistema di controllo agirà sugli indicatori di primo livello e il sistema risponderà secondo le regole intrinseche ad esso ovvero i vincoli di sistema.

- [project management, efficienza del controllo](#)

Tentare di controllare gli indici di primo ordine con gli indici di primo ordine é come voler spostare il cavallo invece di usare le briglie.