Matlab vs. LaTeX

12. januar 2004

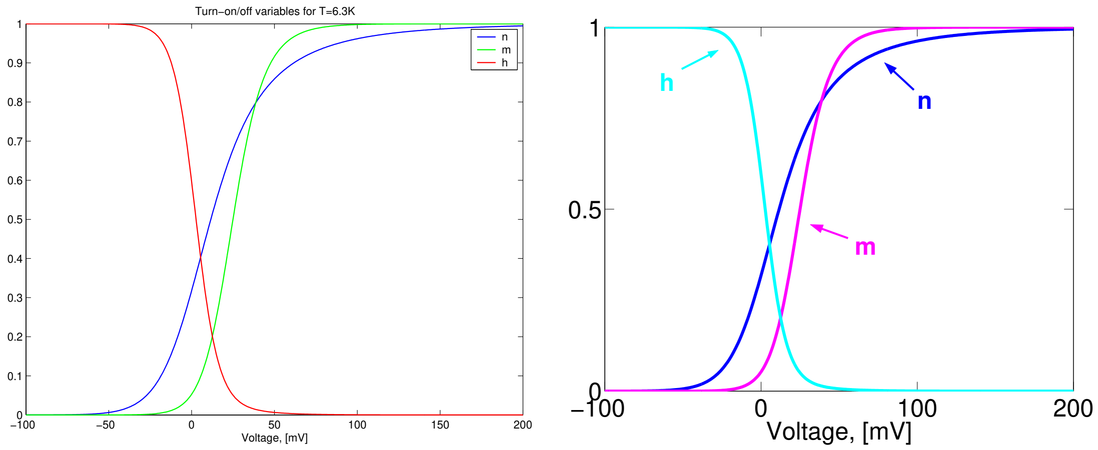
Peter Vingaard Larsen

MatLab og LaTeX er med deres muligheder for henholdsvis at generere og vise EPS[1]–figurer en fantastisk kombination! Der er dog grund til at fremhæve visse ting...

Betragt Fig. 1(a). Her er plottet funktionerne n, m og h som funktion af V med kommandoerne (og se, hvordan man kan benytte parameteren `T` i titlen)

```
plot(V,n,'b', V,m,'g', V,h,'r');
xlabel('Voltage, [mV]');
title(strcat(sprintf('Turn-on/off variables for T=%g',T),'K'));
legend('n','m','h');
```

Læg mærke til, hvordan det er umuligt (nåja, det går måske her, men forestil Jer dette dokument skaleret til A5–format!) at læse akseværdierne, titler, mv.



(a) Simpel udgave. (b) Fancy udgave.

Figur 1: Forskellige muligheder med Matlab.

Se dernæst på Fig. 1(b), der er genereret med kommandoerne

```
fontsize = 24;
axes('Position', [0.15 0.15 .8 0.8])
plot(V,n,'b', V,m,'m', V,h,'c', 'LineWidth', 3);
set(gca,'FontSize',fontsize);
xlabel('Voltage, [mV]');
set(gca,'XTick',[-100 0 100 200])
```

[1] Encapsulated PostScript

```
set(gca,'YTick',[0.0 0.5 1.0])
set(gca,'TickLength', [0.02 0.02]);
text(-65, 0.85, 'h', 'Color', 'c', 'FontWeight', 'bold', 'FontSize', fontsize)
text(100, 0.8, 'n', 'Color', 'b', 'FontWeight', 'bold', 'FontSize', fontsize)
text(60, 0.4, 'm', 'Color', 'm', 'FontWeight', 'bold', 'FontSize', fontsize)
```

Vi lægger mærke til følgende karakteristika:

- På grund af de større typer forsvinder noget af bogstaverne let i .eps–filen med en almindelig figur. Derfor benyttes `axes`–funktionen (for denne og andre funktioner henvises til MatLabs hjælpesider), der mokker selve figuren lidt sammen i vinduet. Derfor er der heller ingen titel i MatLab, men istedet skrives en titel i LaTeX.
- I `plot`–funktionen specificeres en tykkere linie.
- Med `set`–funktionen kan en masse parametre indstilles (som det også blev ovenfor under `plot`). Først er det tekststørrelsen, der får et hak opad, og derefter specificeres det, hvor der skal være akseværdier. Fulgt af en specifikation af disses længde.
- Til sidst indsættes noget tekst i figuren, i stedet for `label`–funktionen. Pilene er lavet i MatLabs figurvindue. Figuren her er i farver, men med passende valgte farver og kurvetykkelse, kan man se forskel på gråtonerne i et S/H–print. Man kan også plotte gråtonerne direkte i MatLab.

Er der brug for mere fifleri giver LaTeX flere muligheder. En pakke ved navn `psfrag` gør det muligt at erstatte tekst i .eps–filen med autoriseret LaTeX–notation. Dermed kan akserne komme til at indeholde komplicerede formler, men der er en masse nusseri forbundet med det, så vi gemmer det til en anden god gang.

Desuden har jeg til denne note brugt pakken `subfigure`, der gør det muligt med individuelle tekster under hver delfigur. Sørg for at inkludere linien

```
\usepackage{subfigure}
```

i dokumentets *preample*, dvs. det, der står før `\begin{document}`. Dernæst er syntaksen:

```
\begin{figure}[H] % [!htb]
  \centerline{
    \subfigure[Subfig 1.]{
      \epsfig{file=fig1.eps, width=7 cm, angle=0} }
    \subfigure[Subfig 2.]{
      \epsfig{file=fig2.eps, width=7 cm, angle=0} } }
  \captionit{Figurtekst.}
  \label{fig:fig1}
\end{figure}
```

God fornøjelse!