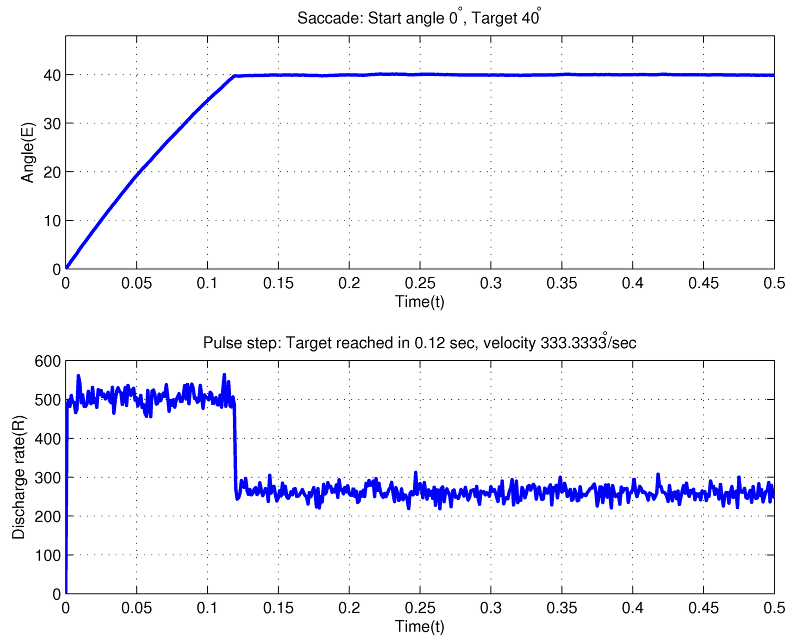
Robotic Eye-Model with Learning of Pulse-step Saccades

Per-Erik Forssén, Bruce Dow, and Dinesh K. Pai, Department of Computer Science, University of British Columbia

Introduction

The human eye shifts gaze on average three times a second. These rapid gaze-shift movements are called *saccades*.

In vivo measurements and current gaze-control models suggest that saccades are implemented using a control strategy known as the *pulse-step model*.



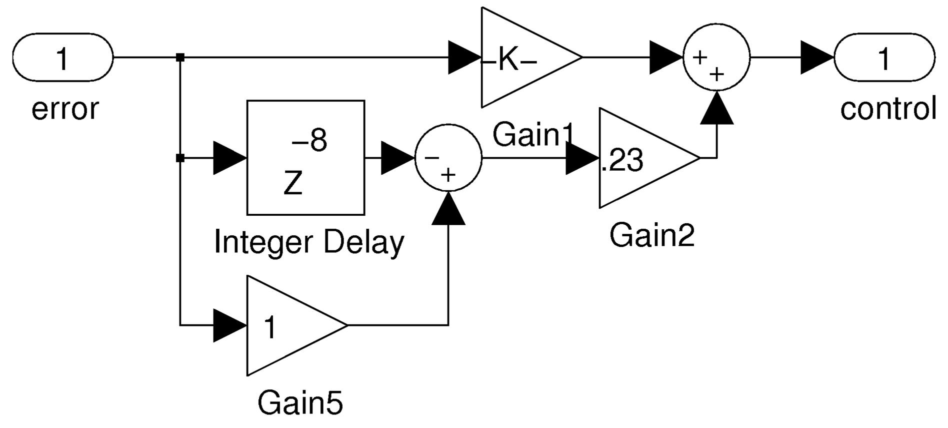
In this model, a maximum discharge rate is exerted initially, and once the target is reached, a resting level rate is maintained.

Human Eye Mechanics Emulator

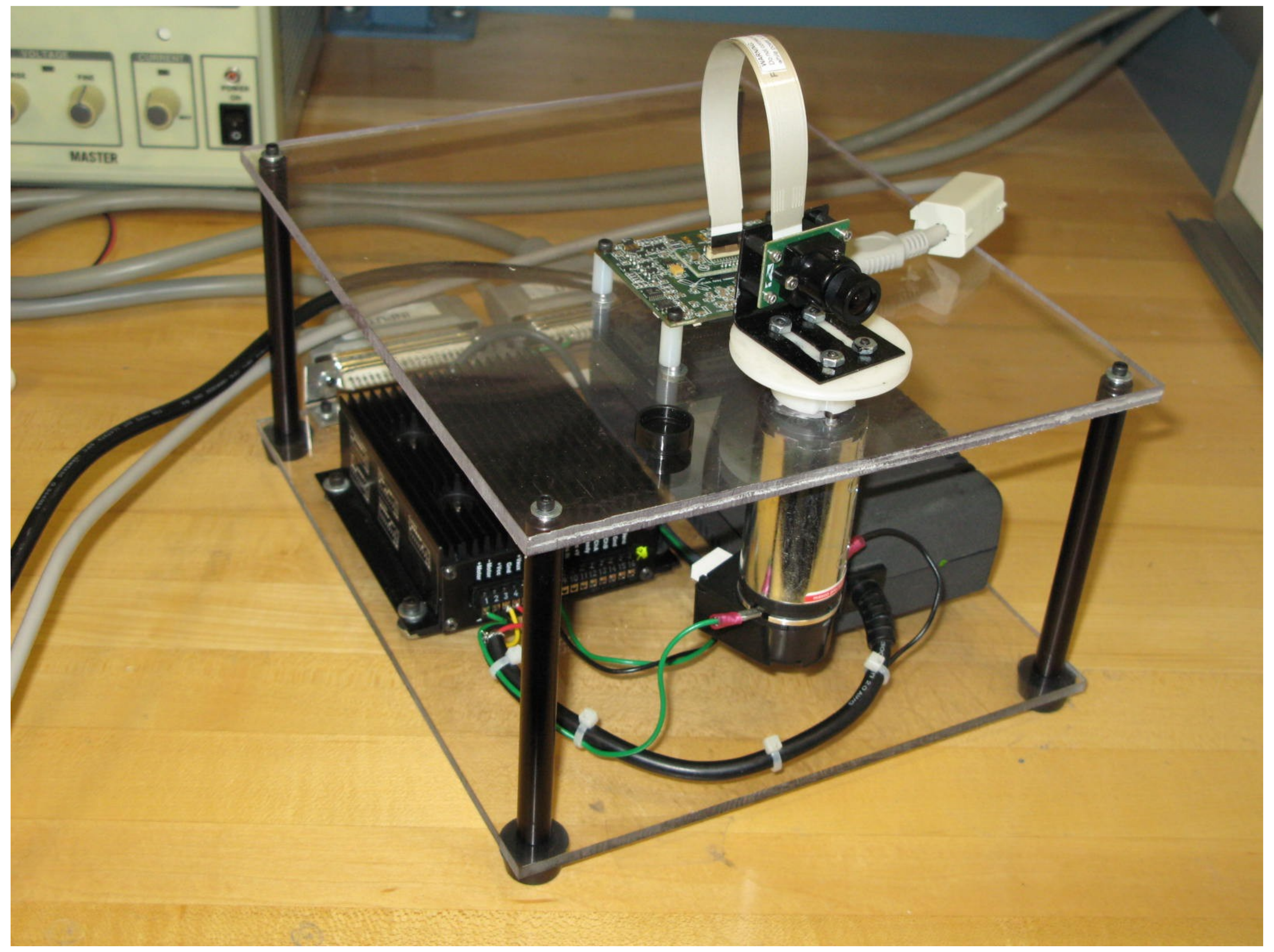
The pulse-step-control model for saccades relies on a compensatory force that drives the eye toward the resting position where it is facing straight ahead.

We have modelled the eye mechanics that provide the resting position using an incremental encoder.

The input to this dynamical model (*error* below) is the output signal from the incremental encoder. Any deviation from encoder value 0° will thus generate a counteracting force and pull the eye back toward its resting position. The model is of PD type (Proportional-Derivative), i.e., the motor voltage y(t) is generated as: $y(t)=k_1x(t)+k_2dx(t)$ (k_1 and k_2 are Gain1 and Gain2 below).



Hardware Design



The physical eye model is based around

- Two plexi-glass platforms, joined together by four black aluminium rods in the corners.
- A Maxon A-max motor. The motor exerts a torque related to supplied voltage.
- A PointGrey Dragonfly2-Extended-Head 1394b camera. The force the motor can exert could potentially allow rotational camera motions of up to 3000°/sec.
- An optical incremental encoder HEDS-5540 A11 from Avago Technologies. This provides angular proprioceptive feedback with a resolution of 0.25°.

The rotational movement of the camera is limited by two bolts to cover a range of 90°, spanning from left 45° to right 45°. The visual output of the camera is transmitted via a thin flat-cable. The cable is quite flexible and the drag it imposes on the rotational movement is quite small. The camera signal is converted to an IEEE1394b signal by the PCB at the other end of the cable, and is transmitted to a Linux PC for further processing.

Learning of the Pulse-step model

This demo demonstrates on-line learning of the *visuomotor function*. For a given image location, **x**, this models the *burst duration*, T_b, and the *resting level* of the motor output, u_s, as a function of the *retinal target disparity*, d=sqrt(|**x**-**c**|²), and the *current discharge level*,

$$(T_b,u_s)=f(d,u)\ .$$

This function can be estimated by performing *probing trial actions* (T_b,u_s), and observing the results, which are the observed retinal displacement, d, and the pulse-step settle time, T_s. The relationship between T_b and u_s can now be learned by trying to *minimise the settle time* using reinforcement learning. To shorten the learning time at the demo, we have here taken this part to be given, and *only learn the burst duration*, T_b, as function of retinal displacement and current encoder value, e.

Action selection

On-line selection of actions for learning is an *active learning* problem. Here we have made use of the observation that the function to be learned is smooth. This allows us to simplify action selection to generation of a uniform coverage of action space [1], while avoiding to bump into the edges.

Non-negative regularisation

Function approximation is performed using *channel associative learning*, where the output is approximated as a weighted sum of channel functions $B_k(e)$:

$$T_b = f(e) = \sum_{k=1}^{K} w_k B_k(e)$$

the kernels $B_k(e)$ are *radial basis functions* with compact support:

$$B_k(e) = \begin{cases} \cos^2((e-e_k)/b) & \text{if } |e-e_k| < b\pi/2 \\ 0 & \text{otherwise} \end{cases}$$

The weights are solved for using a *non-negative constraint* [2]. This constraint serves as a regularisation, that enforces smoothness, and reduces the risk of overfitting.

Demo: Saccade to Letters

Letters written on a whiteboard are detected by *edge filtering* with adaptive thresholding and grouping using morphological filters.

The robot eye moves to a position near the left edge of its range, and successively saccades to each detected letter.

Once a letter is centred, the central portion of the image is extracted and all the pieces are stitched together to form a composite image, hopefully spelling out the written word.

Key References

1. P-E Forssén, *Learning Saccadic Gaze Control via Motion Prediction.* CRV 2007, pp 44-51, Montreal Québec

2. B. Johansson, T. Elfving, V. Kozlov. Y. Censor, P-E Forssén, G. Granlund, *The Application of an Oblique-Projected Landweber Method to a Model of Supervised Learning*, Mathematical and Computer Modelling, vol. 43, April 2006.

3. D. A. Robinson. Control of Eye Movements, In: Brooks, V.B. Editor, 1981. Handbook of physiology, the nervous system.

Acknowledgements

This work was supported in part by the *Swedish Research Council* through a grant for the project Active Exploration of Surroundings and Effectors for Vision Based Robots, the *Canada Research Chairs Program*, *NSERC*, *Canada foundation for innovation*, and *BC KDF*.