

1. In Chapter 6 we noted that the Monte Carlo error can be written as the sum of TD errors (6.6) if the value estimates don't change from step to step. Show that the  $n$ -step error used in (7.2) can also be written as a sum TD errors (again if the value estimates don't change) generalizing the earlier result.

$$\begin{aligned}
 G_{t:t+n} - V(S_t) &= R_{t+1} + \gamma G_{t+1:t+n} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1}) \\
 &= \delta_t + \gamma(G_{t+1:t+n} - V(S_{t+1})) \\
 &= \delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + \cdots + \gamma^{n-1}\delta_{t+n-1} + \gamma^n V(S_{t+n}) - \gamma^n V(S_{t+n}) \\
 &= \sum_{k=t}^{t+n-1} \gamma^{k-t} \delta_k
 \end{aligned}$$

2. (Programming.) With an  $n$ -step method, the value estimates do change from step to step, so an algorithm that used the sum of TD errors (see previous exercise) in place of the error in (7.2) would actually be a slightly different algorithm. Would it be a better algorithm or a worse one? Devise and program a small experiment to answer this question empirically.

There is only a difference in the two methods if we update state estimates for some states from  $\{S_{t+1}, \dots, S_{t+n}\}$  between time steps  $t$  and  $t+n$ . Because of this reason, I would like to try the algorithms on a problem where we may return to the same state shortly. My expectation is that the  $n$ -step update works better, because it uses more recent estimates. Let's try it on a random walk example, like in Example 6.2.