tf.zpk ZPK transfer functions in Matlab

Consider the transfer function:

$$H(s) = \frac{2s+1}{s^2+7s+12} \tag{1}$$
$$= 2\frac{s+1/2}{(s+3)(s+4)}. \tag{2}$$

$z_i = -1/2$

$p_i = -3, -4$

$k = 2$

In the second equality, we have factored the polynomials and expressed them in terms of poles p_i and zeros z_i with terms $(s - p_i)$ and $(s - z_i)$. Note the gain factor 2 that emerges in this form.

sys = zpk([-1/2], [-3, -4], 2)

tf(sys)

Both forms are useful. In the former, two polynomials in s define the transfer function; in the latter, a list of zeros, poles, and a gain constant define the transfer function.

In Matlab, there are two corresponding manners of defining a transfer function. We demonstrate the first, already familiar, method using the `tf` command, which takes polynomial coefficients, as follows.

```
H_tf = tf([2,1],[1,7,12])
```

```
H_tf =

     2 s + 1
  --------------
  s^2 + 7 s + 12

Continuous-time transfer function.
```

Alternatively, we can define the transfer function model with the zpk command using the zeros, poles, and gain constant.

```
H_zpk = zpk([-1/2],[-3,-4],2)
```

```
H_zpk =

   2 (s+0.5)
  -----------
  (s+3) (s+4)

Continuous-time zero/pole/gain model.
```

pole
step

This zpk model will work with all the usual functions `tf` models do. However, if you'd like to convert zpk to `tf`, simply use `tf` as follows.

```
tf(H_zpk)
```

```
ans =

     2 s + 1
  --------------
  s^2 + 7 s + 12

Continuous-time transfer function.
```

Alternatively, we can convert a `tf` model to a zpk model.

```
zpk(H_tf)
```

```
ans =

   2 (s+0.5)
  -----------
  (s+4) (s+3)

Continuous-time zero/pole/gain model.
```