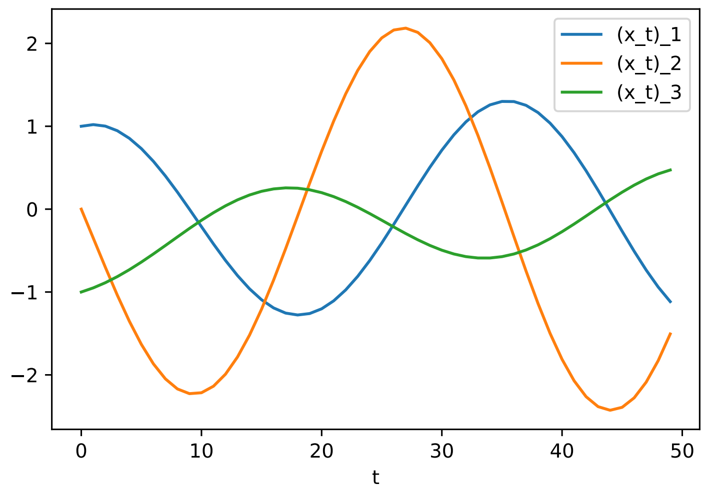

Figure 9.1.: Linear dynamical system simulation.

9.2. Population dynamics

We can create a population dynamics matrix with just one simple line of Python code. The following code predicts the 2020 population distribution in the US using the data of Section 9.2 of VMLS in the `population_data` dataset.

In [ ]:
```
# Import 3 100-vector: population, birth_rate, death_rate
D = population_data()
b = D['birth_rate']
d = D['death_rate']
A = np.vstack([b, np.column_stack([np.diag(1 - d[:-1]),
↪  np.zeros((len(d) - 1))])])
x = D['population']
x = np.power(A,10) @ x
import matplotlib.pyplot as plt
plt.ion()
plt.plot(x)
plt.xlabel('Age')
plt.ylabel('Population (millions)')
plt.show()
```

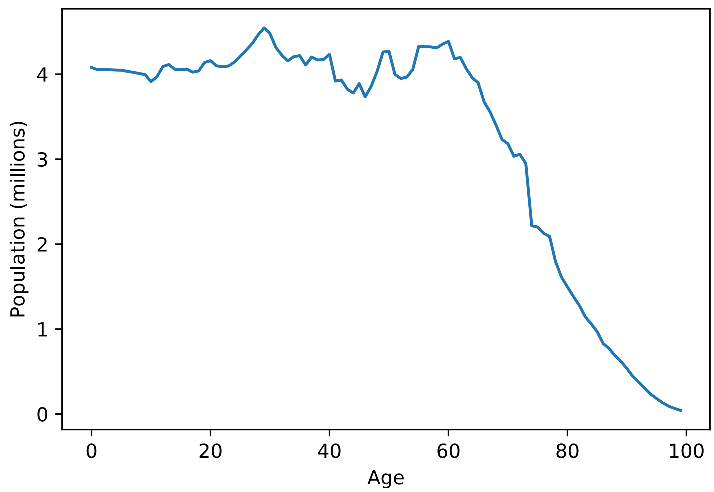

Figure 9.2.: Predicted age distribution in the US in 2020.

9.3. Epidemic dynamics

Let's implement the simulation of the epidemic dynamics from VMLS section 9.3.

In [ ]:
```
T = 210
A = np.array([[0.95,0.04,0,0], [0.05,0.85,0,0], [0,0.1,1,0],
↪  [0,0.01,0,1]])
x_1 = np.array([1,0,0,0])
# state trajectory
state_traj = np.column_stack([x_1, np.zeros((4, T-1))])
# dynamics recursion
for t in range(T-1):
    state_traj[:, t+1] = A @ state_traj[:,t]
import matplotlib.pyplot as plt
plt.ion()
plt.plot(np.arange(T), state_traj.T)
plt.xlabel('Time t')
plt.legend(['Susceptible', 'Infected', 'Recovered', 'Deceased'])
plt.show()
```