

```

classdef ControllerPI < Controller
    properties %Controller State stored here.
        u_operatingpoint %necesario (más o menos) para todos los controladores lineales
        u_integral %estado del integrador
        Kp
        Ki
    end
    methods

        function obj=ControllerPI(Kp,Ki,u_pf)
            arguments
                Kp
                Ki
                u_pf = 0 ; %no input offset if not given
            end
            obj.u_operatingpoint=u_pf; %start at some operation point.
            obj.u_integral=0; %integrator initial condition
            obj.Kp=Kp;
            obj.Ki=Ki;
        end

        function Track(obj,sensor>manual_u,T)
            %do nothing; code is needed for "bumpless transfer".
        end

        function u=ComputeU(obj,setpoint,sensor,T)
            error=setpoint-sensor;
            u=obj.ControlPI(error,T)+obj.u_operatingpoint;
        end

        function u_lin=ControlPI(obj,error,T)
            obj.u_integral= obj.u_integral + obj.Ki*error*T; %Euler Integration
            u_lin = obj.Kp*error + obj.u_integral;
        end

    end
end
end

```

**\*Not valid for "production use": antiwindup and bumpless transfer MUST be incorporated in most applications.**

YouTube Video (Español):

<http://personales.upv.es/asala/YT/V/tank1PIanim.html>