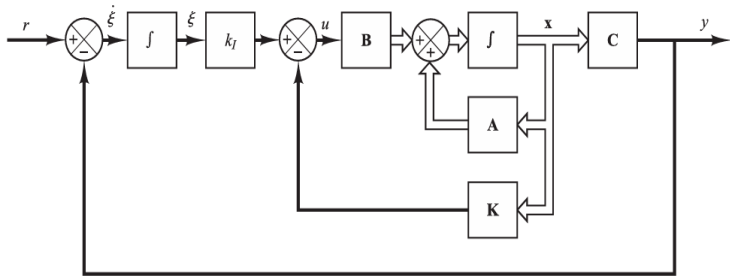
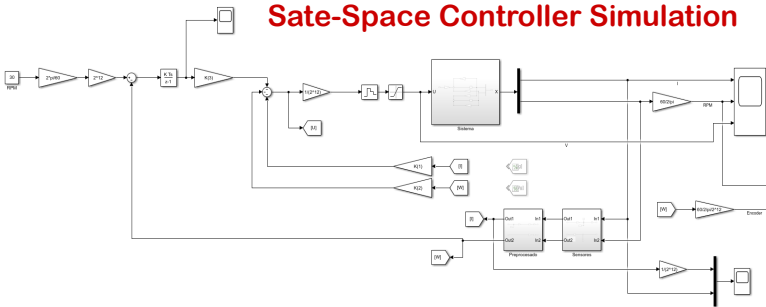


Theoretical State-Space Controller



State-Space Controller Simulation



State-Space Controller Implementation

```

181 void StateSpaceController(control_param_t* param, current_values_t* current, data_shared_t* data)
182 {
183     char zero_r, zero_l;
184     //Check the speed reference, for very low speed impose 0 V output
185     if (data->speed_ref_r == 0){
186         zero_r = 1;
187     } else {
188         zero_r = 0;
189     }
190     if (data->speed_ref_l == 0){
191         zero_l = 1;
192     } else {
193         zero_l = 0;
194     }
195     //Calculates Speed error
196     int error_r = (data->speed_ref_r - data->speed_r);
197     int error_l = (data->speed_ref_l - data->speed_l);
198     //Apply antiwindup saturation action
199     error_r = error_r - 10*param->windup_r;
200     error_l = error_l - 10*param->windup_l;
201     //Integrates the error (Servo System Strategy "Ogata" -> error ~ 0 in steady state)
202     IntegralBackwardEuler(param, error_r, error_l, zero_r, zero_l);
203     //Windup Action
204     IntegralWindup(param, INTEGRAL_SATURATION);
205     //param->x0w_r = param->x0_r;
206     //param->x0w_l = param->x0_l;
207     //Summation of the State Space Controller terms before control action
208     //Summation from Integral part
209     int sum_r = - SignedRightShift(param->K_I*param->x0w_r, 11); //Control parameter was 11 bits left shifted
210     int sum_l = - SignedRightShift(param->K_I*param->x0w_l, 11);
211     //Summation from x2 State (Wheel Speed) feedback
212     sum_r += - SignedRightShift(param->K_speed*data->speed_r, 11); //Control parameter was 11 bits left shifted
213     sum_l += - SignedRightShift(param->K_speed*data->speed_l, 11);
214     //Summation from x1 State (Motor Current) feedback
215     sum_r += - SignedRightShift(param->K_current*current->I_r, 11); //Control parameter was 11 bits left shifted
216     sum_l += - SignedRightShift(param->K_current*current->I_l, 11);
217     //Transforming control action from voltage to PWM (1023/14 ~ 73 -> 0.9% error)
218     //Force the output voltage to be set at 0 V (if the reference speed is 0)
219     if (zero_r)
220     {
221         param->V_r = 0;
222     } else {
223         //Compensating the 12 left bits originally shifted (in current and speed)
224         param->V_r = SignedRightShift(73*sum_r, 12);
225     }
226     if (zero_l)
227     {
228         param->V_l = 0;
229     } else {
230         //Compensating the 12 left bits originally shifted (in current and speed)
231         param->V_l = SignedRightShift(73*sum_l, 12);
232     }
233     //OPEN LOOP
234     /*
235     param->V_r = data->speed_ref_r;
236     param->V_l = data->speed_ref_l;
237     */
238     //It makes the changes to apply the desired voltage in the motors
239     VoltageApplication(param);
240 }

```