Application note

V_{BUS} control algorithm compliant with USB Type-C and Power Delivery specifications

Introduction

The USB Power Delivery specification enables a wide range of devices (e.g., smartphones, laptops, tablets, chargers and adapters, etc.) to select or manage supplied power from 2.5 W up to 100 W over a USB connection, on the basis of a communication protocol that allows negotiating the necessary power according to the application needs.

Up to version 2.0, USB-IF specifications defined the fixed power delivery objects (PDOs) as negotiation goals, overcoming the limit of 5 V in the BUS voltage value (V_{BUS}). Multiple voltage levels (9, 15 and 20 V) have been added to the standard 5 V together with the possibility of defining further fixed voltage values and allowing a wider range of electronic devices to adopt the new protocol to be supplied by a fixed source or a battery-powered device.

With the introduction of the programmable power supply (PPS), PD3.0 specification has included the capability of USB PD power supplies of adjusting their output voltage by 20 mV increments over the advertised range (3.3 - 21 V) and to perform current limiting operation by 50 mA increments.

In this application scenario, V_{BUS} management is crucial for the definition of power solution architectures and the related control algorithms: they have to ensure that voltage levels are stable for any applied load and each augmented power data object (APDO), provided with 20 mV and 50 mA resolutions, is properly provided.

ST has designed a specific algorithm for power source solutions hosting an STM32 microcontroller and compliant with USB Type-C and Power Delivery specifications eligible to feature the PPS.

The algorithm can control the V_{BUS} generated by an AC-DC or a DC-DC converter, satisfying power delivery objects and PPS augmented power data object requests. It also implements the cable drop compensation and minimizes internal drops (i.e., due to R_{DSon} switches).

RELATED LINKS

UM2552: Managing USB power delivery systems with STM32 microcontrollers

AN5225: USB Type-C Power Delivery using STM32 MCUs and MPUs

TA0357: Overview of USB Type-C and Power Delivery technologies

1 Algorithm overview

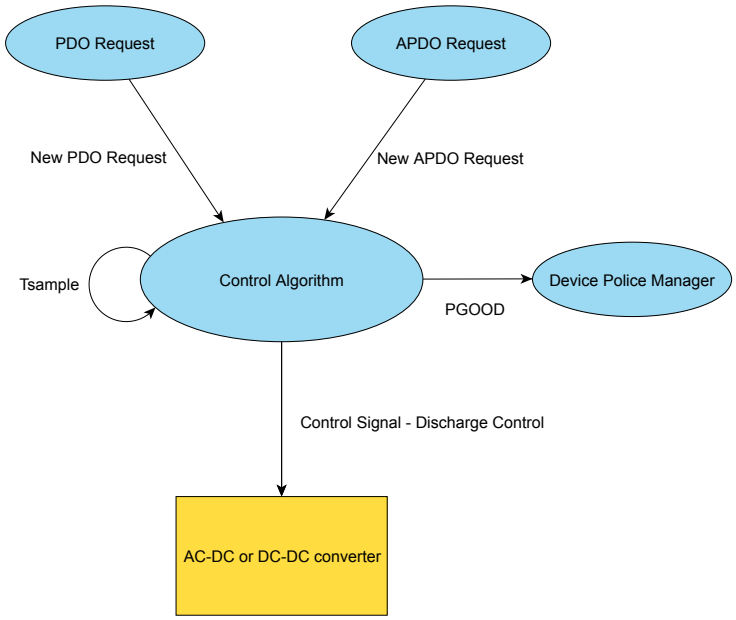
In a power solution, the internal converter represents the key block that manages and characterizes the output voltage. Usually, in such devices, a feedback pin can be driven through an appropriate signal to obtain variable output voltages.

To exploit this converter feature, by associating it with a microcontroller, a specific algorithm has been designed to allow the converter to vary its output voltage in several power solutions, especially the ones supporting the USB Power Delivery specification and negotiate power on the basis of their conditions and capabilities.

Focusing on a USBPD provider solution, the designed algorithm can run on a large range of STM32 and permits the microcontroller to drive a signal on the converter feedback pin, thus allowing the converter to generate an accurate output voltage and fulfill the power requests coming from a sink during a negotiation.

The following figure shows how the algorithm operates in conjunction with the other functional routines and blocks working in a USB Power Delivery solution.

Figure 1. Algorithm operational block diagram



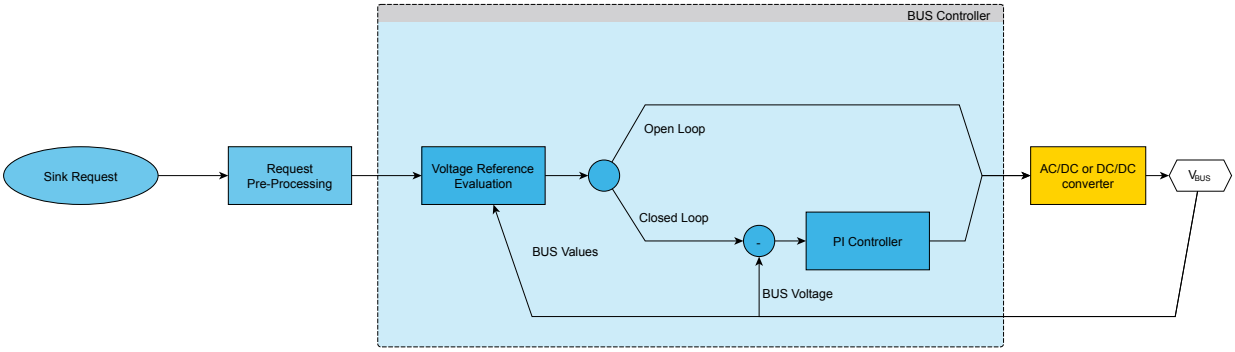
The algorithm operates as a task running each T_{SAMPLE} to permit it to be easily integrated with ST CUBE USBPD Middleware stack and to facilitate the interactions with asynchronous events, such as the incoming requests of new PDOs and APDOs.

When a new PDO or APDO request is received, two appropriate subroutines pre-process the incoming requests and pass the main information to the BUS controller to control the power supply stage (AC-DC or DC-DC) and ensure the output satisfies the request.

The BUS controller core operation is based on a PI controller; in some cases, after the voltage request, an open loop control is activated for few sample times before starting the PI controller to speed up the achievement of the desired voltage in the time requested by the specification (25 ms for voltage steps less than or equal to 500 mV, 275 ms in the other cases).

The BUS controller output is converted into a signal (i.e., PWM, DAC out, etc.) that is connected to the BUS voltage control loop: this signal enables the controller to adapt the V_{BUS} to the requested values.

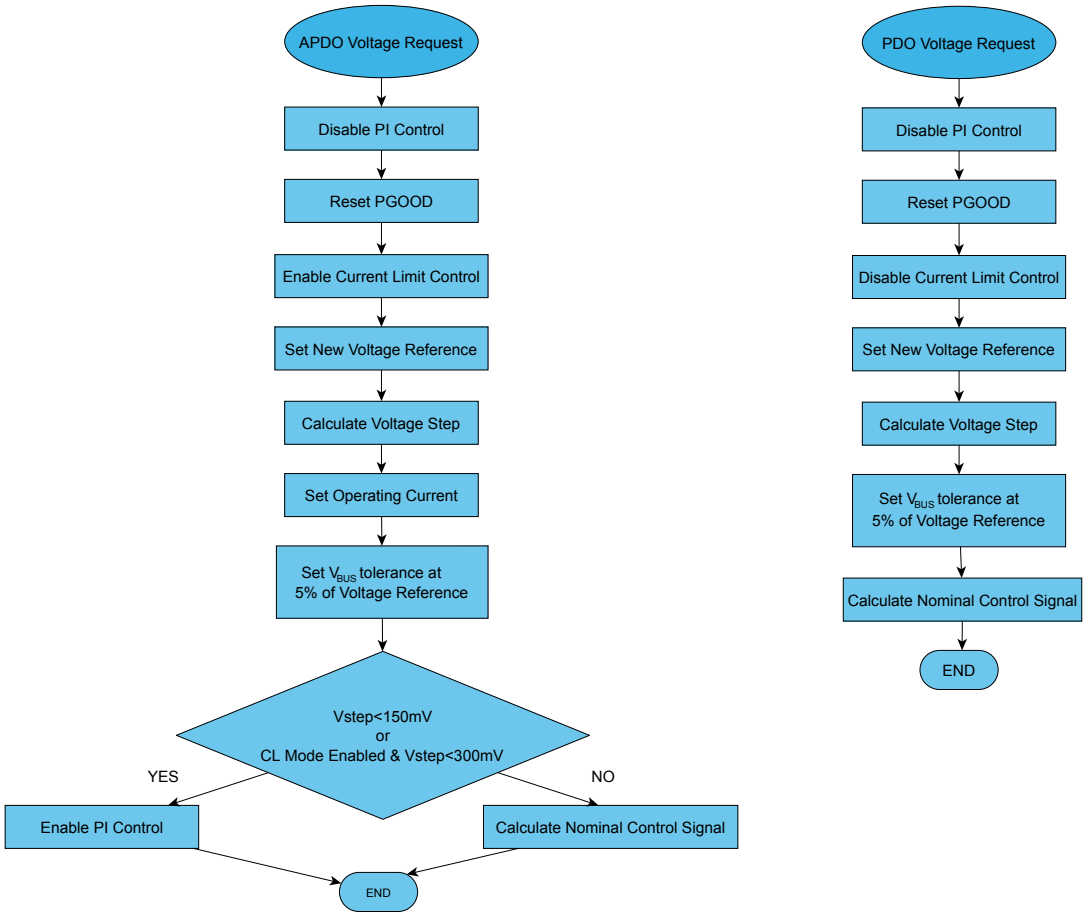
Figure 2. **System overview**



1.1 Sink request management routines

Two main routines manage the new requests coming from an external sink through the DPM layer. These routines use common functions located in the same API library which provide the main data to the BUS controller and properly set some operative parameters.

Figure 3. **APDO and PDO request routine flowchart**



The main functions used to design the two routines are:

- **Disable PI controller** – when a new request arrives, control is immediately deactivated and the procedure to calculate the new set point is initialized
- **Reset PGOOD**– this variable is cleaned every time a new request arrives as the old value is not tolerated anymore
- **Disable/Enable Current Limit Control** – for PDOs, the system does not have to perform current limit control and a current limit flag is reset to allow the control algorithm to disable this feature; for APDOs, instead, the flag is set
- **Set New Voltage Reference** – the control target voltage is set at the new voltage request
- **Calculate Voltage Step** – the voltage step necessary to the system to reach the new required voltage is calculated as the difference between the V_{BUS} current value and the requested voltage in Current Limit mode, otherwise as the difference between the old and the new voltage requests
- **Set V_{BUS} Tolerance** - according to the specification, V_{BUS} tolerance is fixed at 5% of the new requested voltage
- **Set Operating Current** - only for APDO, an operating current is requested; this value is acquired and used by the control algorithm to enter Current Limit mode
- **Calculate Nominal Control Signal** - for both PDO and APDO voltage requests, a signal control nominal value (useful in the open loop control) is calculated to reach the requested value, according to the voltage step and using a table containing the control signal hypothetical value to obtain known values

In APDO requests, when the events described in the conditional block of the flowchart above are verified, the open loop phase is skipped and the PI control is directly enabled.

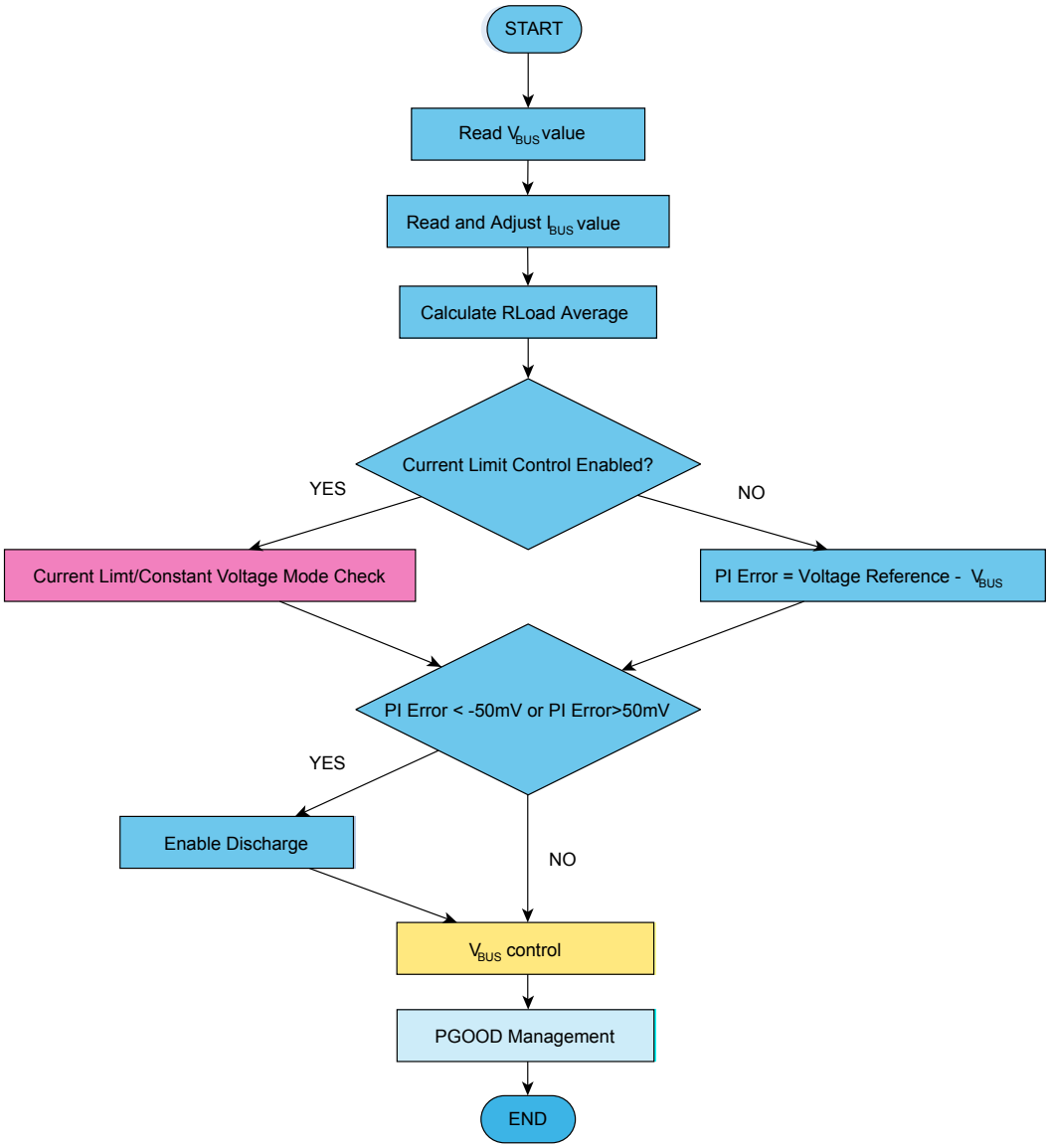
1.2 Bus controller operation

The BUS controller manages the output voltage on the basis of the requested profiles received by the USBPD middleware stack as well as the voltage and current values acquired by the BUS. Hence, it calculates the new load condition (evaluating the equivalent resistance applied by the sink, R_{LOAD}) and generates a control signal applied to the converter feedback pin to correctly set the value of the V_{BUS}. Thus, the controller allows maintaining the set value when in Constant Voltage mode or varying it without overcoming the requested value when operating in Current Limit mode.

The BUS controller minimizes the error between the acquired output voltage V_{BUS} and the desired set point, calculated by the pre-processing routines.

The PGOOD flag rises as soon as the transaction is successfully completed and the V_{BUS} reaches the requested value.

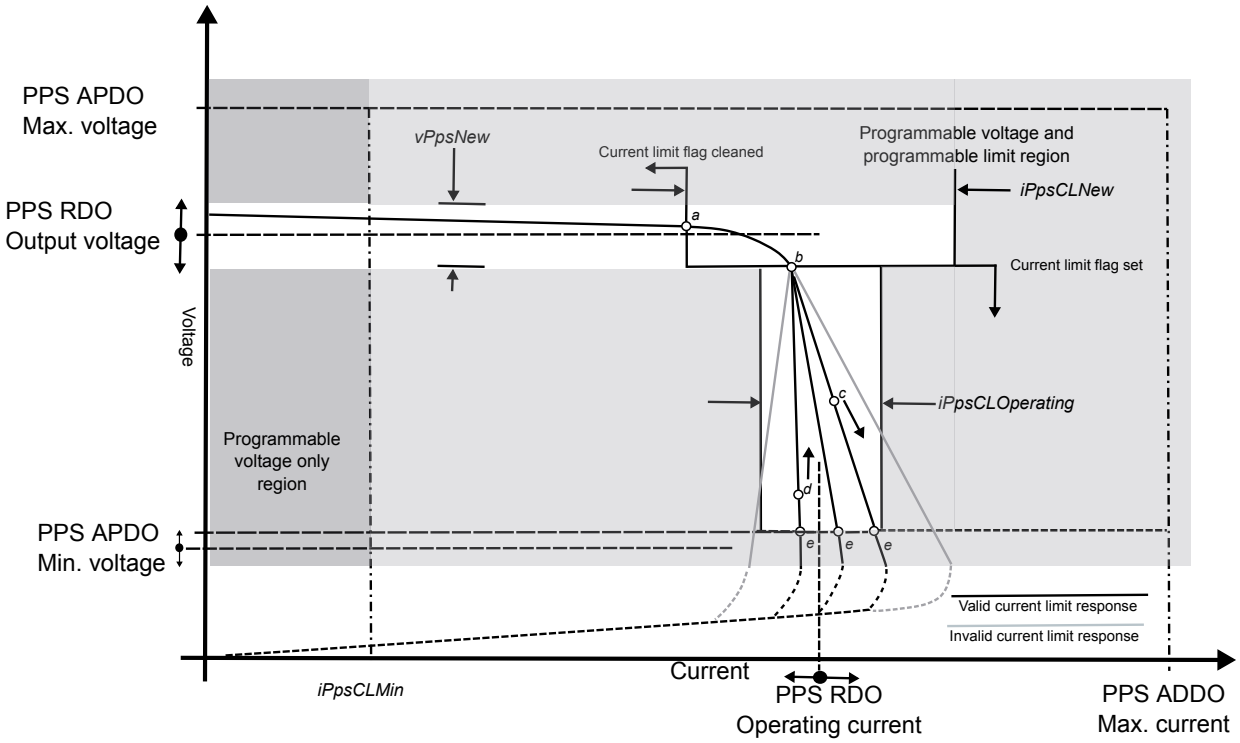
Figure 4. **Bus control task flowchart**



The main BUS controller subroutines are:

- **Current Limit/Constant Voltage Mode Check** that checks whether the system is in the Constant Voltage or in the Current Limit region (see Figure 5) by calculating the voltage reference to maintain the system in the correct operating region. The voltage reference used for the PI could be either the one requested for constant voltage operation or an appropriate value to stay within the current limit region. As the voltage reference is used to evaluate the input error of the PI controller, for PDO requests, this subroutine is disabled and the error is calculated taking into account the PDO voltage request.

Figure 5. PPS programmable voltage and current limit



- **V_{BUS} Control** that manages the control signal to maintain the requested voltage
- **PGOOD Management** that verifies whether the output voltage is in the tolerance range previously calculated

After the acquisition of the BUS voltage and current values, the average resistance of the load, R_{LOAD}, is calculated. The algorithm checks whether the current limit control is enabled or not and then proceeds with the PI error calculation. The current limit control status depends on the request type received.

If a PDO request has been sent, the Current Limit Control is disabled and the input value (error) for the V_{BUS} control routine is calculated as the difference between the requested voltage and the value acquired from the V_{BUS}.

For APDO requests, the Current Limit Control routine is enabled and the error value is directly managed by the Current Limit/Constant Voltage Mode, as described in Section 2.1 .

In both cases, the error value is the input of the V_{BUS} control routine. If the calculated error is over ±50 mV, the control routine enables the discharge mechanism to let the system reach the due voltage value faster.

The PGOOD Management routine checks whether the voltage is in the tolerance range, previously calculated, and releases a confirmation flag.

2 Bus controller blocks

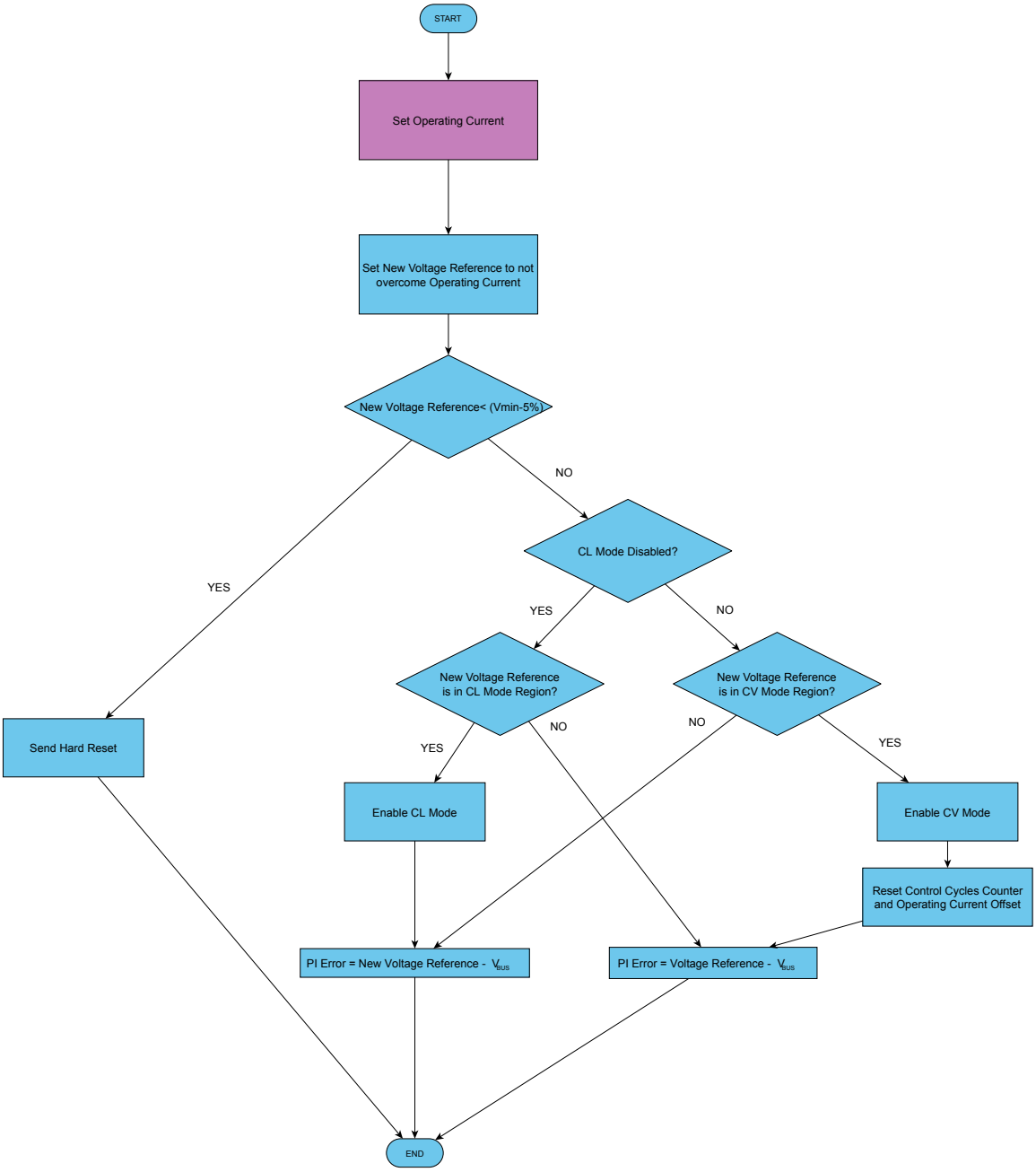
2.1 Current limit/constant voltage mode check

This subroutine recognizes the requested operative mode (Current Limit (CL) or Constant Voltage (CV)) and drives the controller to set the right control strategy.

During this subroutine, an operating current is requested as current limit and its value is maintained even if the load conditions vary, eventually decreasing the output voltage to limit the output current.

By evaluating the load resistance, the subroutine calculates the maximum output voltage to not overcome the operating current: if this value is lower than the requested value and inside the CL region (see Figure 5. PPS programmable voltage and current limit), CL mode is enabled and the voltage reference is set to this calculated value, otherwise the voltage reference is set to the requested voltage and CV mode is enabled.

Figure 6. Current limit control algorithm flowchart



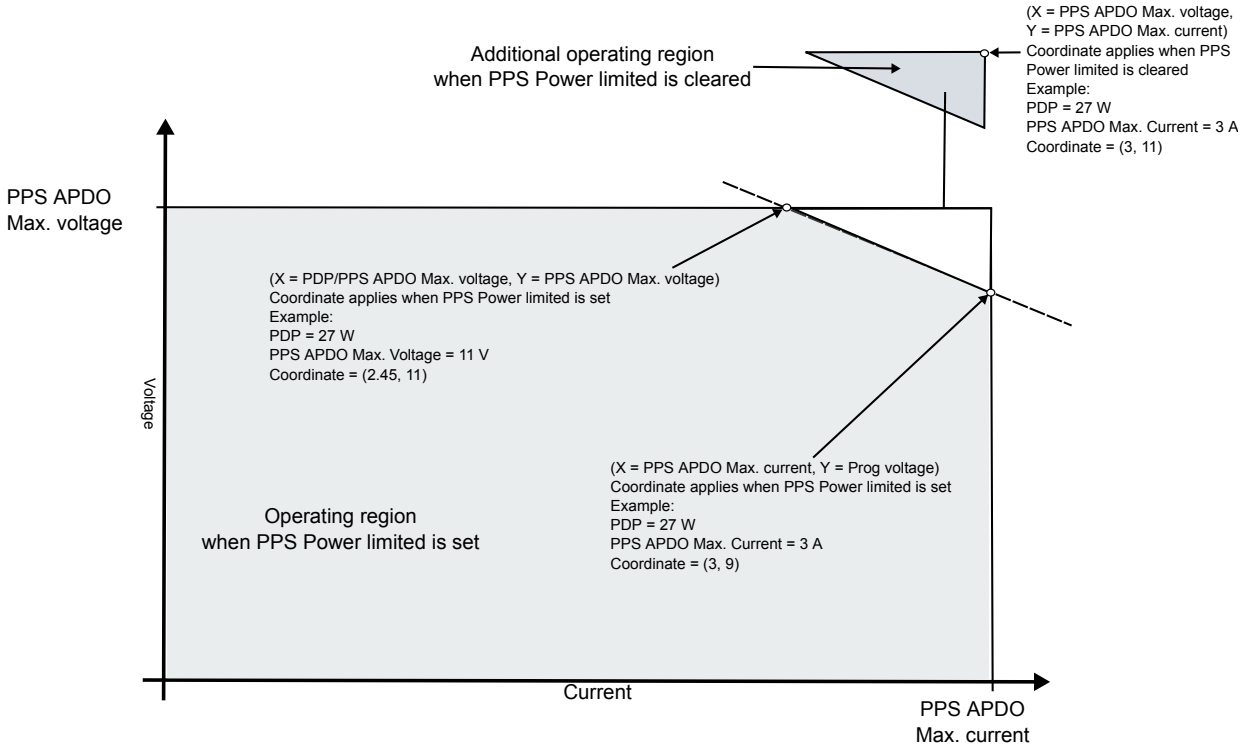
In Current Limit mode, the new calculated voltage reference can be under the minimum output voltage minus 5%; in this case, a Hard Reset is sent as it is impossible to maintain the operating current at the output.

2.1.1 Setting the operating current

An APDO request contains information about the operating current. The operating current value must be changed for power limitation and load resistance variation.

For power limited systems, if a voltage request exceeds the maximum possible power, the operating current is decreased and set to remain inside the requested power profile (see Figure 8).

Figure 7. PPS power limit operating regions

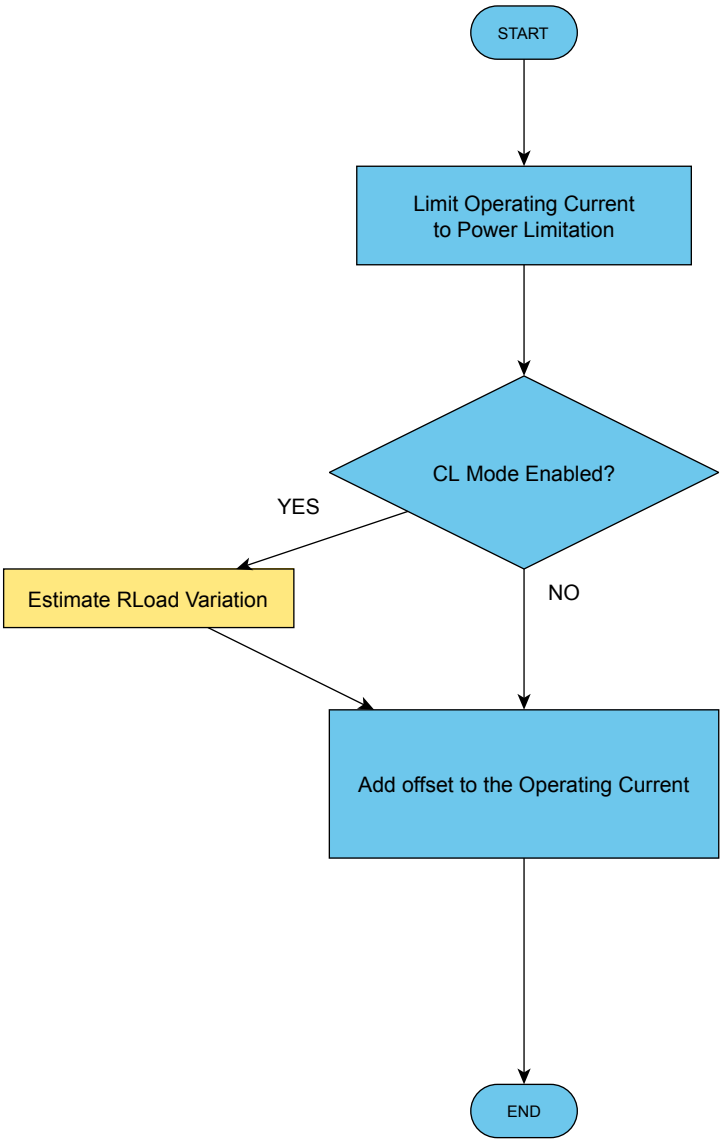


As stated in the USB-PD standard, when the load resistance decreases, the output current should stay the same or increase slightly and, as the load resistance increases, the output current should stay the same or decrease slightly.

When the system is in CL mode, a slight offset (positive in case of decrease or negative in case of increase) is added to the operating current.

To estimate an increase or decrease of the load resistance a specific subroutine has been implemented.

Figure 8. **Operating current setting algorithm flowchart**

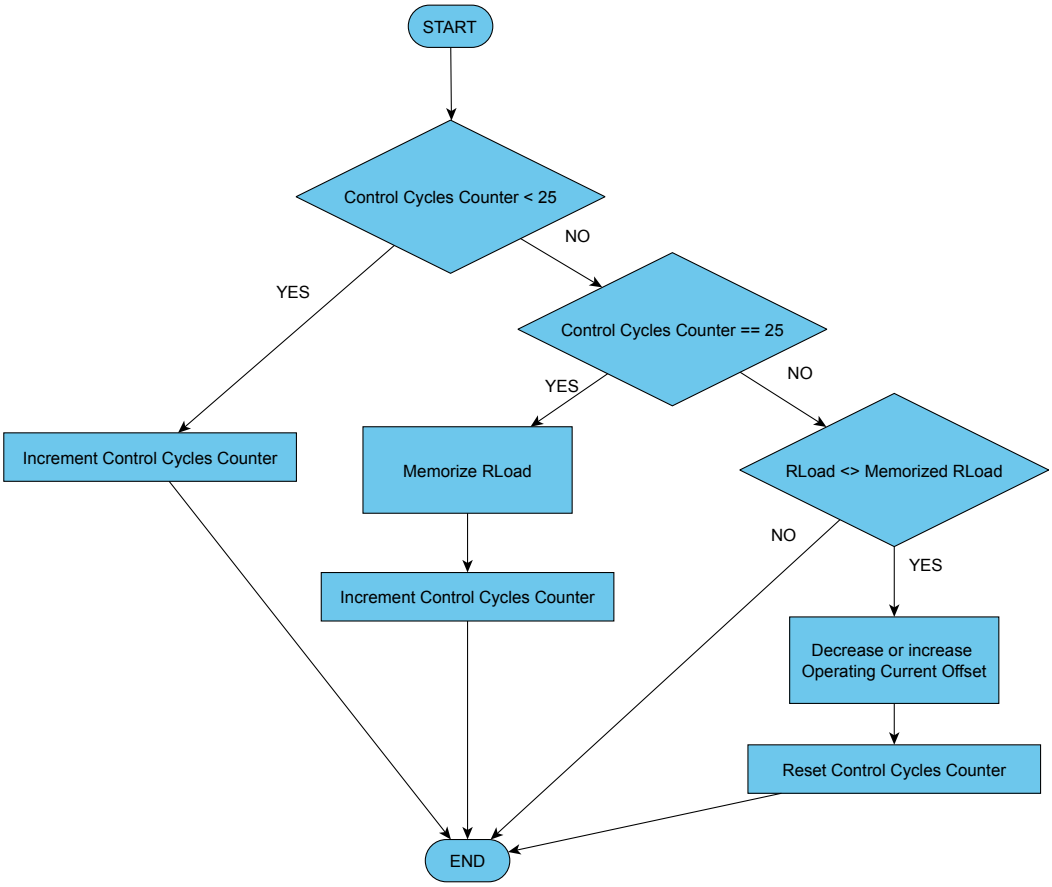


2.1.2 R_{Load} variation estimation

This subroutine estimates whether a consistent variation of the load has occurred. If so, an offset is added to the operating current to avoid undesired increases or decreases in the output current.

At the beginning or after a resistance variation, the algorithm waits for approximately 50 ms to ensure the load is stable, then acquires and saves the load resistance value which is compared with the new acquired value. If there is a consistent variation of the load, an offset is added to the operating current and the new load resistance value is saved.

Figure 9. R_{Load} variation estimation flowchart



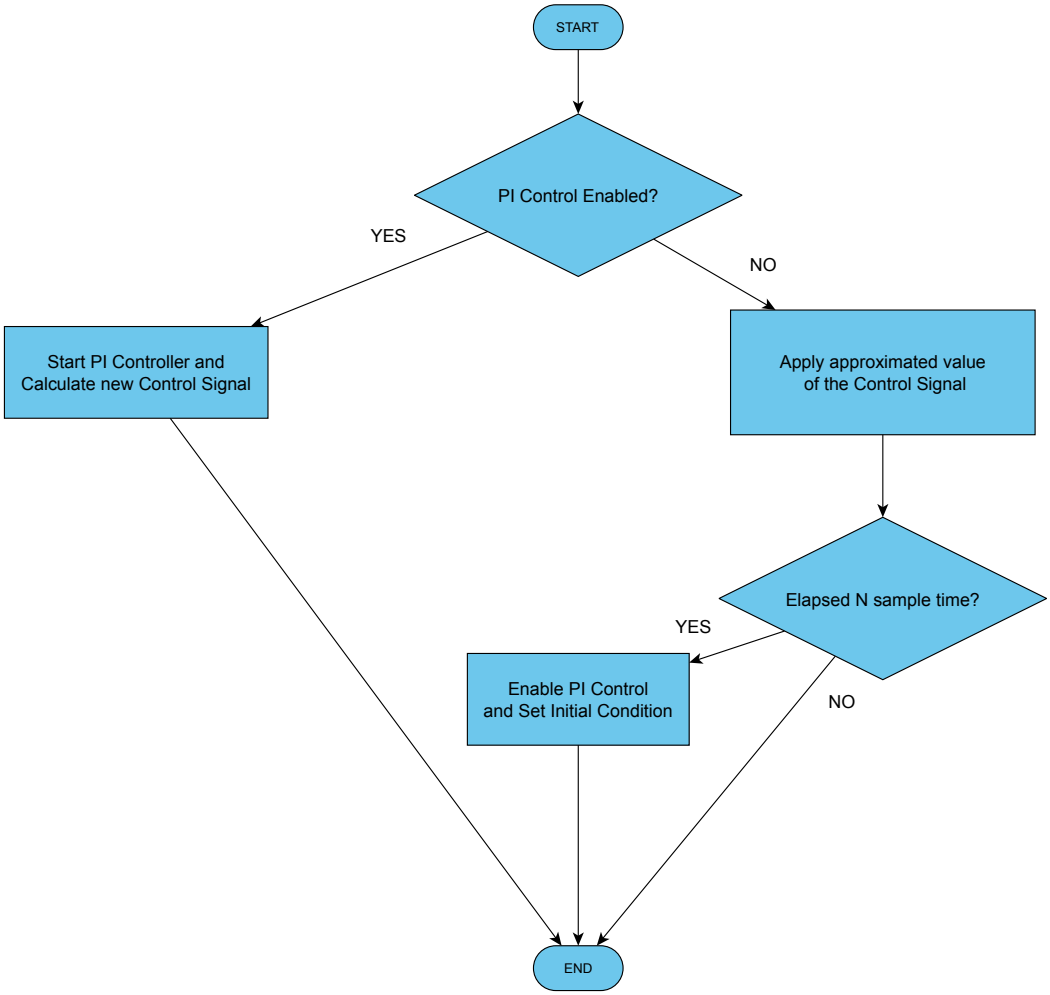
2.2 V_{BUS} control

This subroutine reduces the error (estimated in mV), that is the difference between the voltage reference (requested voltage or calculated voltage) and the voltage read on the bus.

When a new request is accepted by the PDO/APDO request routines, the PI control inside the voltage control routine is disabled, except when, for APDO, the voltage step required is less than 150 mV or 300 mV in Current Limit mode (see Section 1.1 Sink request management routines).

In the open loop phase, the control signal nominal value calculated in the PDO/APDO request routine is directly applied to the control for a number of sample times directly proportional to the voltage step. After these steps, the PI control is enabled to achieve the desired voltage and maintain the constant value even under load variations.

Figure 10. **Voltage control routine flowchart**



2.3 PGOOD management

This routine manages the PGOOD flag.

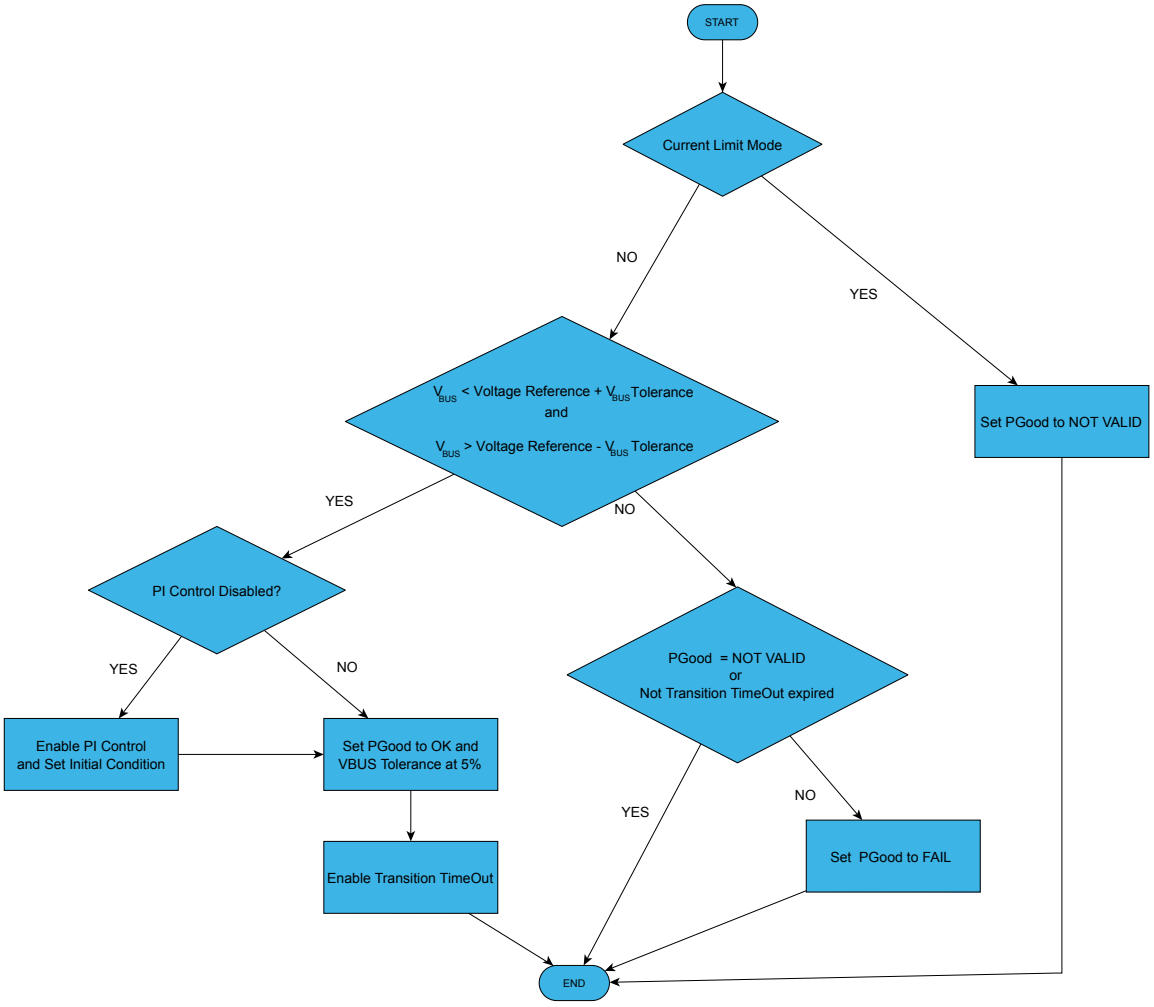
The PGOOD flag can assume three different states: OK, FAIL or NOT VALID.

The last state is activated at the beginning of a transition and when the system is in Current Limit mode. When a new request arrives, the system is not in error if it reaches the desired value in time; thus, at the beginning of a transition, the flag is set to NOT VALID. Moreover, as in Current Limit mode the V_{BUS} does not follow the required voltage to not exceed the operating current, it would be outside the tolerance but not in error, so, also in this case, the flag is set to NOT VALID.

When the V_{BUS} reaches the requested value, the flag is set to OK and the routine does not change the flag state for the maximum duration of a transition as stated in the specifications. In fact, during a transition the V_{BUS} can trespass the tolerance threshold but not be in error (refer to USBPD specifications).

If the flag is not in NOT VALID state or it is not in a transition and the V_{BUS} is outside the tolerance, the PGOOD flag is set to FAIL.

Figure 11. **PGOOD management routine flowchart**



3 Algorithm parameters

Table 1. List of parameters to be set for the correct algorithm operation

Name	Description	Block reference
PPS_CONTROL_DELAY	Control algorithm sample time (ms)	BUS controller
PPS_CONTROL_DISABLED_COUNT	Default number of sample times for the open loop control operation	APDO and PDO requests
PPS_CONTROL_DISABLED_COUNT_MAX	Maximum number of sample times for the open loop control operation	APDO request
PPS_CONTROL_WAIT	Number of sample times to wait for a stable measurement of the load resistance	Set operating current
PPS_MOBILE_AVG_ORDER	Order of the mobile average to measure the load resistance	BUS controller
PPS_IBUS_CC_THRESHOLD	I_{BUS} tolerance (mA)	BUS controller
PPS_VBUS_DELTA_ENABLER	Minimum voltage step ($\|V_{BUS} - V_{Request}\|$) below which the PI control (mV) is directly enabled	APDO request
PPS_VBUS_DELTA_OMF	In Current Limit mode, the minimum voltage step ($\|V_{BUS} - V_{Request}\|$) below which the PI control (mV) is enabled	APDO request
PPS_VBUS_TOLLERANCE_PERC	V_{BUS} tolerance percentage	APDO request
PPS_VBUS_UV_THRESHOLD	Undervoltage V_{BUS} threshold (mV)	BUS controller
PPS_VBUS_UV_THRESHOLD_VALIDITY	Minimum voltage value to be considered as a valid measurement	BUS controller
PPS_VBUS_OV_THRESHOLD	Overvoltage V_{BUS} threshold (mV)	BUS controller
PPS_VREF_NOMINAL	Nominal V_{REF} (mV) (default 5000 mV)	BUS controller
PPS_POWER_NOMINAL	Nominal power (mW)	Set operating current
PPS_POWER_TOLERANCE	Power tolerance (mW)	Set operating current
PPS_PI_KP	PI control K_P parameter	BUS controller
PPS_PI_KI	PI control K_I parameter	BUS controller
PPS_PI_LOWERLIMIT	PI control minimum output value	BUS controller
PPS_PI_UPPERLIMIT	PI control maximum output value	BUS controller
PPS_DUTY_MAX	Maximum duty cycle value	BUS controller
PPS_DUTY_5V_NOMINAL	Nominal duty cycle for 5 V output	PDO request
PPS_DUTY_9V_NOMINAL	Nominal duty cycle for 9 V output	PDO request
PPS_DUTY_STEP_VALUE	V_{BUS} step (mV) used to create the duty cycle look-up table	BUS controller APDO request
PPS_DUTY_STEP_COUNT	Number of elements on the control signal look-up table	BUS controller APDO request

4 Conclusions

The implementation of the V_{BUS} control algorithm applies appropriate control strategies to drive a power converter (AC-DC or DC-DC) with programmable power supply (PPS) complying with the USB-PD specifications for Power Data Objects (PDOs) and Augmented Power Data Objects (APDOs).

The algorithm key features are:

- BUS control with 20 mV (for voltage) and 50 mA (for current) granularity
- operating region (Constant Voltage, Current limit) identification
- voltage transition management to meet requested timing

The V_{BUS} is managed by a PI controller that offers a high level of flexibility to adapt the algorithm performance by varying a restricted set of characteristic parameters

This controller has been tested with the STEVAL-USBPD27S and with a series of USB compliance tools obtaining excellent results in line with the specifications.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
28-Oct-2020	1	Initial release.

Contents

List of tables

List of figures